

David Goll

**Usability Improvements durch Recommendations in  
einem Online-Portal**

**DIPLOMARBEIT**

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fakultät: Elektro- und Informationstechnik

Mittweida, 10. Januar 2012

David Goll

# Usability Improvements durch Recommendations in einem Online-Portal

eingereicht als

## DIPLOMARBEIT

an der

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fakultät: Elektro- und Informationstechnik

Mittweida, 10. Januar 2012

Betreuer: Prof. Dr.-Ing. Andreas Ittner

Dipl.-Inf. (FH) Gordon Tschirner

Vorgelegte Arbeit wurde verteidigt am:

Bibliographische Beschreibung:

David Goll: Usability Improvements durch Recommendations in einem Online-Portal.  
- 2012. - 71 S.

Mittweida, Hochschule Mittweida, Fakultät Elektro- und Informationstechnik, Diplomarbeit, 2012

# Inhaltsverzeichnis

Abkürzungsverzeichnis	v
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
Listingverzeichnis	viii
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen</b>	<b>3</b>
2.1 Usability . . . . .	3
2.1.1 Erklärung und Definition von Usability . . . . .	4
2.1.1.1 Definition nach ISO-Norm 9241 . . . . .	4
2.1.1.2 Definition nach Nielsen . . . . .	7
2.1.2 Fazit . . . . .	10
2.2 Online-Portal . . . . .	11
2.3 Matchmaking . . . . .	14
2.4 Recommendations . . . . .	15
2.5 A/B-Test . . . . .	16
<b>3 Implementierung</b>	<b>19</b>
3.1 Matchmaking . . . . .	19
3.1.1 Konzept . . . . .	19
3.1.2 Implementierung . . . . .	24
3.1.2.1 Views-Filter-Handler . . . . .	25
3.1.2.2 Matchmaking-Einstellungsmaske . . . . .	31

3.1.3	Fazit . . . . .	34
3.2	Recommendations . . . . .	35
3.2.1	Konzept . . . . .	36
3.2.2	Implementierung . . . . .	39
3.2.2.1	Konfiguration des A3S-Moduls . . . . .	39
3.2.2.2	HTML-Template-Anpassungen . . . . .	39
3.2.2.3	Datenexport . . . . .	46
3.2.2.4	Erster Filter . . . . .	49
3.2.2.5	Ordnersicherung mittels .htaccess . . . . .	50
3.2.2.6	Zweiter Filter . . . . .	52
3.2.3	Fazit . . . . .	56
3.3	A/B-Test . . . . .	56
3.3.1	Konzept . . . . .	56
3.3.2	Implementierung . . . . .	57
3.3.2.1	Konfiguration des user_splitting-Moduls . . . . .	57
3.3.2.2	Implementierung der Funktionalität . . . . .	59
3.3.2.3	Verwendung in Panels . . . . .	63
3.3.2.4	Auswertung mittels QlikView . . . . .	65
3.3.3	Fazit . . . . .	68
<b>4</b>	<b>Schlusswort</b>	<b>70</b>
<b>A</b>	<b>Anhang</b>	<b>I</b>
A.1	Tabellen . . . . .	I
A.2	Inhalt der CD . . . . .	III
	<b>Glossar</b>	<b>IV</b>
	<b>Literaturverzeichnis</b>	<b>VIII</b>
	<b>Erklärung selbstständiger Arbeit</b>	<b>XIII</b>

# Abkürzungsverzeichnis

AOL .....	America Online
API .....	Application Programming Interface
CCK .....	Content Construction Kit
CMS .....	Content Management System
CSS .....	Cascading Style Sheets
CSV .....	Comma Separated Values
CTR .....	Click Through Rate
DIN .....	Deutsches Institut für Normung
HTML .....	Hypertext Markup Language
ISO .....	International Organization for Standardization
PHP .....	PHP: Hypertext Preprocessor
SQL .....	Structured Query Language

# Abbildungsverzeichnis

2.1	AOL-Software (Version von 1997) . . . . .	11
3.1	Von Views bereitgestellte Filter-Handler . . . . .	23
3.2	Views-Einstellungsmaske: Filter hinzufügen . . . . .	31
3.3	Views-Einstellungsmaske: Feld hinzufügen . . . . .	32
3.4	Views-Einstellungsmaske: Operator hinzufügen . . . . .	32
3.5	Views-Einstellungsmaske: Referenz-Feld hinzufügen . . . . .	33
3.6	Matchmaking-Einstellungsmaske . . . . .	34
3.7	Anzeige von Recommendations mit Standard-HTML-Template . . . . .	38
3.8	Anzeige von Recommendations und Tooltip mit angepasstem HTML- Template . . . . .	44
3.9	Ablauf der Filterung vor der Anzeige der Empfehlungen . . . . .	45
3.10	Struktur der CSV-Datei . . . . .	46
3.11	Felder der Data-Export-View . . . . .	47
3.12	Filter der Data-Export-View . . . . .	50
3.13	user_splitting-Einstellungsmaske . . . . .	58
3.14	Recommendations-Region des Panels . . . . .	63
3.15	Visibility-Rule des Recommendation-Blocks 0 . . . . .	64
3.16	QlikView: Übersicht über die A/B-Testverteilung . . . . .	65
3.17	QlikView: Übersicht über die Gastfamilien des A/B-Testes . . . . .	66
3.18	QlikView: Übersicht über die Variante 0 des A/B-Testes . . . . .	67
3.19	QlikView: Übersicht über die Variante 1 des A/B-Testes . . . . .	67
3.20	QlikView: Übersicht über die gesendeten Kontaktanfragen der Variante 0 des A/B-Testes . . . . .	68
3.21	QlikView: Übersicht über die gesendeten Kontaktanfragen der Variante 1 des A/B-Testes . . . . .	68

# Tabellenverzeichnis

3.1	Gegenüberstellung System-Feldname zu Quelltext-Feldname . . . . .	41
A.1	Key - Wertgegenüberstellung der Geschlechtsangaben . . . . .	I
A.2	Key - Wertgegenüberstellung der Landesangaben . . . . .	II
A.3	Gastfamilien-Felder des Recommendation-Templates und deren Ver- wendung im System . . . . .	III



# Listingverzeichnis

3.1	Matchmaking-Anbindung an die Views-API . . . . .	24
3.2	Definition eines neuen Filter-Handlers . . . . .	25
3.3	Views-Hook für die Filter-Handler-Verwendung . . . . .	26
3.4	Definition eines neuen Filter-Operators . . . . .	27
3.5	op_cck-Methode Teil 1: Speichern von Benutzerdaten in das \$value_list- Array . . . . .	28
3.6	op_cck-Methode Teil 2: Speichern der \$ignore_value in das \$value_list- Array . . . . .	28
3.7	op_cck-Methode Teil 3: SQL-Statement mit Werten des \$value_list- Arrays erzeugen . . . . .	29
3.8	Beispiel-Wert für die Variable \$sql_string . . . . .	30
3.9	Beispiel-Wert für die Variable \$sql_string nach Überprüfung der «ignore_value» . . . . .	30
3.10	Benutzer-Felder laden . . . . .	40
3.11	Benutzer-Felder im Recommendation-Template benutzen . . . . .	41
3.12	CSS: span-Tag verstecken . . . . .	43
3.13	CSS: Recommendation-Anzeige formatieren . . . . .	43
3.14	Cronjob . . . . .	48
3.15	.htaccess-Datei . . . . .	51
3.16	.htpasswd-Datei . . . . .	51
3.17	Von A3S bereitgestellter Hook zum Filtern der Empfehlungen . . . . .	53
3.18	Unterscheidung der Benutzertypen . . . . .	54
3.19	Benutzer auf inaktiv überprüfen . . . . .	54
3.20	Benutzer auf männlich überprüfen . . . . .	55
3.21	Löschen nicht erlaubter Recommendations . . . . .	55

3.22 Benutzer bei Login überprüfen . . . . .	59
3.23 Benutzer in Testgruppen einteilen . . . . .	60
3.24 Benutzergruppe überprüfen . . . . .	62
3.25 Benutzergruppe aus Datenbank auslesen . . . . .	62

# 1 Einleitung

Viele Web-Designer legen ausschließlich Wert auf das Aussehen einer Website und lassen die Bedienbarkeit dieser völlig außer Acht. Benutzer, die sich im Internet bewegen, betrachten die besuchte Website aber nicht genau, sondern «scannen» diese auf der Suche nach den gewünschten Informationen. Wenn die Benutzer nicht gleich das Gesuchte finden, wird eine Suchmaschine aufgerufen und eine andere Website besucht.

Es wird in dieser Arbeit jedoch nicht darauf eingegangen, wie sich mit einem guten Website-Layout die Usability verbessern lässt oder ob ein Menü links oder rechts dargestellt werden soll. Es wird vielmehr versucht, die Usability zu erhöhen, indem der Benutzer durch so wenig wie möglich Interaktion mit der Website an die für ihn passende Information kommt. Denn gerade online ist es für den Betreiber einer Website nicht nachzuvollziehen, welche Kenntnisse der Benutzer, der gerade Gast auf seiner Website ist, im Umgang mit dem World Wide Web hat.

Ziel dieser Arbeit ist die Erhöhung der Usability eines Online-Portales. Um dies zu erreichen, sollen Wege aufgezeigt werden, mit denen einem Benutzer in einem Online-Portal zielgerichtete Informationen zur Verfügung gestellt werden können. Außerdem sollen dem Benutzer weiterführende Informationen angezeigt werden, auf die er eventuell nicht gekommen wäre. Ein zusätzliches Ziel ist es, eine Möglichkeit zu schaffen, womit diese Wege nach dem Implementieren evaluiert, ausgewertet und gegebenenfalls überarbeitet werden können.

In dieser Arbeit wird die Einblendung von «Recommendations» realisiert, wodurch versucht wird, die Usability eines Online-Portales so weit zu erhöhen, dass Benutzer intuitiv zu den für sie wichtigen Informationen kommen. Außerdem wird – wie bei

etablierten Online-Partner-Vermittlungen – ein System entwickelt, welches die zur Verfügung stehenden Benutzerdaten verwendet. Somit wird eine Suche realisiert, die ohne erneute Eingabe des Benutzers passende Ergebnisse liefert. Dies bezeichnet man im Allgemeinen als «Matchmaking».

Abschließend dienen «A/B-Tests» als Testumgebung, um zwei Varianten der berechneten Empfehlungen miteinander vergleichen zu können. Diese soll dabei helfen, die Empfehlungen weiter zu verbessern und demzufolge den Benutzer noch mehr bei seiner Suche nach Informationen zu unterstützen.

## 2 Grundlagen

### 2.1 Usability

Im deutschsprachigen Raum war noch vor zwei Jahrzehnten der Begriff «Usability» weitgehend unbekannt (vgl. Richter und Flückiger, 2007, 3). „Eine eher kleine Fachgemeinde von «Software-Ergonomen» befasste sich in dieser Zeit mit der «Gebrauchstauglichkeit» bei der Verwendung von interaktiven Systemen“ (ebd.). Zunächst beschäftigten sich die Software-Ergonomen mit der Untersuchung der Mensch-Computer-Interaktion (ebd. 7f.). Wesentliches Ziel dabei war die Unterstützung des Anwenders bei dem Gebrauch der Software, sodass er seine Arbeitsaufgaben effektiv, effizient und zufriedenstellend erledigen kann (vgl. Dahm, 2005, 38).

Mit dem Einsetzen des Internet-Booms Ende der 1990er Jahre setzten viele Unternehmen auf das World Wide Web (kurz www) (vgl. Richter und Flückiger, 2007, 8). Es ist den Firmen nun möglich, zunächst sich selbst und später ihre (eventuell selbst hergestellten) Waren weltweit zu präsentieren. Die Ware kann darüber hinaus ohne einen Zwischenhändler direkt in einem Online-Shop zum Kauf angeboten werden. Die Nachfrage nach der Erstellung von benutzerfreundlichen Websites steigerte sich enorm und der Begriff «Web-Usability» wurde zu einem neuen Schlagwort (ebd.). Heute ist die Usability – also die Benutzerfreundlichkeit – einer Website ein ausschlaggebender Bestandteil, der den Erfolg einer Website bestimmt. Ein Benutzer muss die Website bedienen und effizient nutzen können (vgl. Gube, 2011, erster Abschnitt, übersetzt von DG). Derzeit bietet fast jedes vierte deutsche Unternehmen (23%) seine Produkte oder Dienstleistungen auf einer Online-Präsenz im Internet an (vgl. Statistisches Bundesamt, 2010). Wenn ein Benutzer diese nicht bedienen kann,

wird er vermutlich bei der Konkurrenz kaufen (wenn deren Online-Präsenz besser ist).

Im Folgenden soll der Begriff «Usability» mit Hilfe von zwei verschiedenen Ansätzen geklärt werden.

### 2.1.1 Erklärung und Definition von Usability

Es gibt viele Definitionen und Übersetzungen für Usability. Hansjörg Rampl nähert sich dem Begriff beispielsweise über den englischen Wortstamm. Er übersetzt «to use» mit *benutzen, gebrauchen, verwenden* und «the ability» mit *die Möglichkeit* (Rampl, 2007). Usability übersetzt Hansjörg Rampl daher mit «die Möglichkeit zu benutzen» (ebd.). Eine andere Übersetzung, die auch in Normen und Fachbüchern verwendet wird, ist «Gebrauchstauglichkeit» (vgl. dazu ebd.; vgl. Heinecke, 2005).

Manchmal wird Usability aber lediglich als Gütekriterium für die Gestaltung einer Benutzeroberfläche verstanden. Die Kriterien dafür sind dann beispielsweise die Anordnung der Bedienelemente, die Verständlichkeit der angezeigten Bezeichnungen oder Dialoge und die Anzahl der notwendigen Klicks für das Erreichen des Ziels. Der Begriff ist jedoch umfassender (vgl. dazu Richter und Flückiger, 2007, 3): „Software-Anwendungen oder Produkte weisen eine hohe Usability auf, wenn sie von den vorgesehenen Benutzern einfach erlernt und effizient verwendet werden können und diese damit ihre beabsichtigten Ziele und Aufgaben zufriedenstellend ausführen können“ (ebd.).

Eine oft referenzierte Definition für Usability ist die ISO-Norm 9241 (vgl. ebd., 4), auf die nun genauer eingegangen wird.

#### 2.1.1.1 Definition nach ISO-Norm 9241

1995 wurde die Norm 9241 von der *International Organization for Standardization* zunächst unter dem Namen «Ergonomische Anforderungen für Bürotätigkeiten mit

Bildschirmgeräten» eingeführt. 2006 wurde der ISO-Titel in «Ergonomie der Mensch-System-Interaktion» geändert, um die Einschränkung auf Büroarbeiten aufzuheben (vgl. Rampl, 2007). Die ISO-Norm besteht aus 17 Teilen und enthält im Teil 10<sup>1</sup> die «Grundsätze zur Dialoggestaltung». Dieser Teil beschreibt die ergonomische Gestaltung – unabhängig von einer bestimmten Dialogtechnik – von Software (Europäisches Komitee für Normung, 1995).

Teil 10 umfasst die sieben folgenden Grundsätze für die Gestaltung und Bewertung eines Dialoges, welche als wichtig erkannt worden sind (ebd., 4):

- Aufgabenangemessenheit: Der Benutzer soll insofern bei seiner Arbeitsaufgabe unterstützt werden, dass er sie effektiv und effizient erledigen kann (vgl. ebd., 5f.).
- Selbstbeschreibungsfähigkeit: Jeder Dialogschritt soll mittels Rückmeldung des Dialogsystems – oder durch Anfrage des Benutzers – verständlich erklärt sein (vgl. ebd., 6ff.).
- Steuerbarkeit: Nachdem ein Benutzer den Dialog gestartet hat, soll er in der Lage sein, die Richtung und Geschwindigkeit dieses Dialogs bis zum Erreichen des Zieles zu beeinflussen (vgl. ebd., 8f.).
- Erwartungskonformität: Der Dialog soll konsistent sein und den Merkmalen des Benutzers entsprechen. Damit sind allgemeine Kenntnisse und Erfahrungen des Benutzers gemeint, die hier mit einfließen sollen (vgl. ebd., 9f.).
- Fehlertoleranz: Das Arbeitsergebnis soll trotz erkennbarer fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand durch den Benutzer erreicht werden (vgl. ebd., 10f.).
- Individualisierbarkeit: Lässt das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe zu, so ist dies individualisierbar (vgl. ebd., 11f.).

---

<sup>1</sup>Dieser Teil wurde 2006 durch Teil 110 ersetzt, die Grundsätze jedoch blieben gleich.

- Lernförderlichkeit: Das Dialogsystem selbst soll den Benutzer beim Erlernen des Dialogsystems unterstützen und anleiten (vgl. Europäisches Komitee für Normung, 1995, 12f.).

Diese Grundsätze gelten als allgemeine Leitlinien (vgl. ebd., 4). Die Umsetzung dieser Grundsätze in einem Dialogsystem ist abhängig von dem jeweiligen Benutzer (für den das Dialogsystem gedacht ist), den Arbeitsaufgaben, der Arbeitsumgebung sowie der jeweils genutzten Dialogtechnik (vgl. ebd.).

In der ISO-Norm ist oft die Rede von «Dialog» und «Benutzer». Um die Begriffe zu standardisieren, werden auch diese in der Norm definiert. Die Norm erläutert *Dialog* und *Benutzer* wie folgt (ebd., 5):

1. Dialog: Eine Interaktion zwischen einem Benutzer und einem Dialogsystem, um ein bestimmtes Ziel zu erreichen.
2. Benutzer: Ein Mensch, der mit dem Dialogsystem arbeitet.

Die Entwickler gingen beim Erstellen der ISO-Norm auf die Belange der Benutzer ein. Das Anwenden dieser Norm soll zu Benutzerschnittstellen führen, die gebrauchstauglicher und konsistenter sind. Damit ermöglicht man dem eigentlich Profitierenden dieser Norm – dem Benutzer – eine höhere Produktivität (vgl. ebd., 4). Es sollte beim Entwickeln eines Systems auf die Grundsätze dieser Norm eingegangen werden. Denn für ein optimales Design des Dialogs ist ihre Einhaltung ein wesentlicher Erfolgsfaktor (vgl. Völkel, 2002).

Zu beachten ist aber, dass die Grundsätze abhängig voneinander sind. In manchen Fällen wird es daher nicht möglich sein, alle Grundsätze im gleichen Umfang umzusetzen. Hierbei ist es zu empfehlen, die Vorteile des einen Grundsatzes mit den Vorteilen eines anderen abzuwägen (vgl. Europäisches Komitee für Normung, 1995, 4). „Die Anwendbarkeit und die jeweilige Wichtigkeit hängen vom speziellen Anwendungsfall, von den Benutzergruppen und von der gewählten Dialogtechnik ab.“ (ebd.) Damit wird die bestmögliche Usability für den Benutzer gewährleistet.



Neben der ISO-Norm als Definition für Usability sollte auch Nielsen erwähnt werden, welcher sich ausgiebig mit diesem Thema befasst hat.<sup>2</sup>

### 2.1.1.2 Definition nach Nielsen

Nach Jakob Nielsen (vgl. Nielsen, 1994, 23ff.) ist es wichtig, die Usability nicht nur als eindimensionale Eigenschaft zu sehen, sondern vielmehr als eine Gesamtheit von mehreren Komponenten. Er stützt sich dabei auf fünf Attribute (ebd.):

- die Erlernbarkeit,
- die Effizienz,
- die Merkbarekeit,
- die Fehlertoleranz
- und die Zufriedenheit.

Nielsen vertieft somit die Grundsätze aus der ISO-Norm (vgl. hierzu Abschnitt 2.1.1.1). Er geht nicht darauf ein, wie ein Dialog aufgabengerecht gestaltet sein könnte, sondern legt sein Augenmerk vielmehr auf den Benutzer. Nachdem ein Dialog nach der ISO-Norm erstellt worden ist, sollte man diesen nach den fünf Attributen von Nielsen noch einmal evaluieren und gegebenenfalls anpassen. Damit kann das Qualitätsmerkmal Usability enorm gesteigert werden. Was genau unter einem Qualitätsmerkmal zu verstehen ist, erläutert das folgende Zitat:

„Usability ist ein Qualitätsmerkmal, wie einfach etwas zu benutzen ist. Es geht genauer gesagt darum, wie schnell Menschen die Benutzung eines Gegenstands erlernen können, wie effizient sie während seiner Benutzung sind, wie leicht sie sich diese merken können, wie fehleranfällig der Gegenstand ist und wie er den Nutzern

---

<sup>2</sup>Laut Richter und Flückiger ist Nielsens Buch «Usability Engineering» ein Meilenstein für die Entstehung des Fachgebiets Usability Engineering (vgl. Richter und Flückiger, 2007, 7f.).

gefällt. Wenn die Nutzer einen Gegenstand weder nutzen möchten noch können, bräuchte er eigentlich gar nicht zu existieren.“ (Loranger und Nielsen, 2008, xvi)

Nielsen stützt sich bei seiner Evaluierung der Usability auf fünf Attribute. Nachstehend werden diese näher betrachtet.

### *Die fünf Usability-Attribute nach Nielsen*

Mit Hilfe der fünf Attribute schätzt Nielsen die Qualität der Usability eines Dialoges ein (vgl. Nielsen, 1994, 23ff.). Ein Benutzer soll demnach die höchstmögliche Zufriedenheit durch ein einfaches Erreichen seines Zieles erlangen. Mit «Erlernbarkeit» meint Nielsen, dass das System leicht verständlich sein soll, um sofort mit der Arbeit beginnen zu können. Nachdem das System einmal erlernt worden ist, soll es möglich sein, dies auch sehr «effizient» zu nutzen. Um einem Nutzer das erneute Erlernen des Systems zu ersparen, ist der Punkt der «Merkbarkeit» ein weiteres Attribut. Wenn ein Benutzer das System einmal erlernt hat und es für seine Arbeit nutzt, kann die «Zufriedenheit» aufgrund ständig auftretender Fehler leiden; deshalb muss an einer «Fehlertoleranz» gearbeitet werden.

Im Folgenden soll anhand der Erhebung der einzelnen Attribute genauer auf diese eingegangen werden. Nielsen geht in seinem Buch «Usability Engineering» auf verschiedene Wege ein, mit denen sich diese Attribute erheben und messen lassen (vgl. dazu ebd., 29ff.):

Die «Erlernbarkeit» ist seiner Meinung nach das am einfachsten zu messende Attribut. Eine Kennzahl für diese lässt sich erheben, indem Benutzer, die das System nicht kennen, eine Aufgabe mit diesem durchführen müssen. Ist diese Aufgabe zufriedenstellend erledigt worden, gilt das System als erlernt. Um eventuell aussagekräftigere Ergebnisse zu erzielen, ist es möglich, die Benutzer auch mehrere Test-Aufgaben erledigen zu lassen. Dabei wird die benötigte Zeit gemessen (vgl. dazu ebd., 29).

Dass Nielsen mit «Effizienz» nicht nur das Erreichen eines definierten Zieles meint, wird bei seinem verwendeten Test-Ansatz deutlich. Hat ein Benutzer, der das System vorher nicht kannte, die gestellte Aufgabe zufriedenstellend absolviert, so nutzt er das System effizient. Nielsen verwendet für das Testen der Effizienz allerdings

auch Benutzer, die sich bereits im System auskennen und misst die Zeit, die für das Erreichen des Zieles benötigt wurde (vgl. Nielsen, 1994, 31). Dieser Ansatz verdeutlicht, dass Nielsen das System auf eventuelle «Shortcuts»<sup>3</sup> oder – für erfahrene Benutzer – vereinfachte Eingabemöglichkeiten untersucht.

An den Tests der «Merkbarkeit» eines Systems nehmen nur Benutzer, die das System bereits kennen, teil. Es kann zum einen direkt nach dem Verwenden des Systems über einen Fragebogen festgestellt werden, wie sich die Abläufe bei den Benutzern manifestiert haben. Zum anderen ist es aber auch möglich, die Test-Benutzer nach einer gewissen Zeit, in der sie das System *nicht* verwenden dürfen, nochmals dieselben Tests durchführen zu lassen. Nielsen favorisiert diese zweite Variante (vgl. dazu ebd., 32) und bezieht sich damit auf eine Studie von Mayes (Mayes u. a., 1988). Diese Studie besagt, dass sich Benutzer, die gerade noch an einem System gearbeitet haben, danach an keinerlei Details erinnern können. Wenn sich diese Benutzer jedoch wieder vor das System setzen, können sie ohne Probleme mit diesem umgehen (vgl. dazu Nielsen, 1994, 32). Eine Evaluierung mit Fragebögen wäre demnach wenig aussagekräftig.

Mit «Fehlertoleranz» meint Nielsen lediglich die Anzahl der Fehler, die beim Verwenden des Computer-Systems auftreten können (vgl. ebd., 32f.) und so gering wie möglich gehalten werden sollten. Um diese Anforderung einzuhalten, könnte man sich ein System vorstellen, welches fehlerhafte Eingaben selbst berichtigt und somit den Benutzer entlastet. Mit diesem Ansatz kann die Usability enorm gesteigert werden.

Das Attribut, welches am schwersten mit einem geeigneten Wert versehen werden kann, ist die «Zufriedenheit». In einem ersten Ansatz wurde die Zeit gemessen, die ein Benutzer zum Erfüllen der Test-Aufgaben braucht. Wenn diese Zeit sehr gering war, konnte von einer hohen Zufriedenheit ausgegangen werden, da der Benutzer schnell das vorgegebene Ziel erreichte. Diese Tatsache ist allerdings nicht allgemeingültig, da sich dieser Aspekt mit dem Einzug der Computer-Systeme in die Heim-, Spiele- und Unterhaltungselektronik geändert hat (Virzi, 1991 nach Nielsen, 1994, 33). Ein Benutzer ist demnach bei Unterhaltungselektronik nur dann zufriedenzustellen, wenn

---

<sup>3</sup>Sind Tastenkombinationen, mit denen bestimmte Programmfunktionen schneller ausgeführt werden können.

er bei der Benutzung relativ lange braucht. Das liegt daran, dass bei dieser Art von Computer-Systemen der Unterhaltungswert für den Benutzer wichtiger ist als das schnelle Erreichen seines Zieles (Carroll und Thomas, 1988 nach Nielsen, 1994, 33). Somit wiesen Carroll und Thomas bereits 1988 darauf hin, dass der Fokus auf Usability in der ziel- und aufgabenorientierten Ausrichtung nicht ausreiche, und dass Spaß und Freude bei der Nutzung interaktiver Produkte berücksichtigt werden müsse (vgl. Burmester u. a., 2001, 33). Da ihrer Meinung nach die Berücksichtigung der Zeit allein nicht ausreichte, wurden zur Messung von Zufriedenheit psychophysiologische Tests durchgeführt. Bei diesen Tests wurden unter anderem Pupillendehnung, Hauttemperatur, Blutdruck, Herzfrequenz und Adrenalin Spiegel während der Verwendung des Systems gemessen (vgl. dazu Nielsen, 1994, 34). Diese Art der Datenerfassung war aber für die meisten Software-Entwicklungen aufgrund des hohen Aufwandes undenkbar. Nielsen favorisiert hier, die Zufriedenheit – wenn auch sehr subjektiv – durch einen einfachen Fragebogen festzustellen (vgl. ebd.). Je mehr Benutzer an dieser Befragung teilnehmen, desto aussagekräftiger wird auch das Ergebnis dieses Tests (vgl. dazu ebd.).

### 2.1.2 Fazit

Zusammenfassend ist zu sagen, dass es seit Beginn von Computer-Systemen das Bestreben nach besserer Usability gibt. Ein Benutzer soll sich in dem Computer-System zurechtfinden und es (für sich) zufriedenstellend benutzen können. Eine DIN-Norm gibt die allgemeinen Grundsätze an, wie Dialoge in einem System gestaltet werden sollten, damit der Benutzer intuitiv mit seiner Arbeit beginnen kann. Mit Hilfe einer geeigneten Form kann man das Wirken der Norm in den Dialogen des Computer-Systems erheben. Wie diese Formen aussehen können und was damit letztendlich beschrieben wird, stellt Nielsen mit seinen fünf Attributen der Usability dar.

## 2.2 Online-Portal

Mit der Verbreitung des Internets nahm auch das Angebot an global verfügbaren Informationen zu. Viele Benutzer waren zum einen wegen mangelnder Kenntnis über das Medium und zum anderen wegen der so schnell wachsenden Komplexität überfordert. Dies machte eine übersichtliche Strukturierung und Bündelung der Informationen erforderlich. Somit entstanden die ersten Portale, die als „Einstiegsseiten in das Internet“ (Kirchhof u. a., 2004, 3) galten. Portale gliedern die Informationen nach Themen und geben dem Benutzer die Möglichkeit, mit Hilfe einer Suchfunktionalität zielgerichtet an die gesuchten Informationen zu kommen. Bekannte Beispiele sind AOL oder Yahoo<sup>4</sup> (vgl. ebd., 3). Die Abbildung 2.1 zeigt das Startfenster der AOL-Software aus dem Jahr 1997:



Abbildung 2.1: AOL-Software (Version von 1997)

Dieses Fenster wurde dem Benutzer direkt nach der Einwahl in das Internet angezeigt. Der Benutzer kann hier zwischen 12 verschiedenen Themen wählen und bekommt

<sup>4</sup>Im Rahmen dieser Arbeit soll nur auf AOL eingegangen werden.

damit einen zielgerichteten Einstieg in die zahlreichen Informationen des Internets. Auf der linken Seite des Fensters bekommt der Benutzer Zugang zu weiteren Internetdiensten wie E-Mail, Chat und World Wide Web. Auch eine Suche steht dem Benutzer in der unteren Leiste des Fensters zur Verfügung («Finden»).

Heute wird nach wie vor bei der Verwendung der AOL-Software eine Website aufgerufen, die Informationen nach Themen gliedert und eine Suchmöglichkeit bietet.

Da die Online-Portale von Yahoo und AOL für jeden Benutzer zugänglich sind und immer noch als Einstiegsseiten in das Internet verstanden werden, ist die Informationsvielfalt auf diesen Seiten enorm.

Die Definition für «Portal» von Steve Williams zeigt im Vergleich dazu, dass ein Portal auch als Einstieg in eine bestimmte inhaltliche Richtung verwendet werden kann: „A portal can be described as an 'anchor' or starting point that makes all the types of information (destinations) available to a designated audience by passing through the one point.“ (Williams, 2011).

Das Zitat zeigt, dass Steve Williams unter einem Online-Portal eine Website versteht, die an ein bestimmtes Publikum adressiert ist und ihm demnach zielgerichtete Informationen zur Verfügung stellt. Somit muss ein Online-Portal nicht immer vielfältige Themengebiete präsentieren.

Das Online-Portal, auf das in dieser Arbeit eingegangen werden soll, bietet Au-pair und Gastfamilien die Möglichkeit, sich zu finden. Bevor darauf weiter eingegangen werden soll, wird ein Exkurs zum Thema «Drupal» eingefügt, da das Portal auf diesem System basiert.

Drupal ist ein sehr ausgereiftes, flexibles, auf PHP basierendes, datenbankbasiertes CMS (vgl. Graf, 2008, 37; vgl. Drupal, 2011). Da es ausschließlich Inhalte auf einem Webserver verwaltet, fällt Drupal in die Gattung der «Web Content Management Systeme» (vgl. Graf, 2008, 37). Aufgrund der Modularität von Drupal lassen sich die unterschiedlichsten Anpassungen schnell und effizient durch Module realisieren (vgl. Drupal, 2011). Module sind Dateien, die in PHP geschrieben sind und Funktionalitäten beinhalten, die Drupal verwenden kann. In anderen CMS werden Module meist als «extensions» bezeichnet (vgl. Robbins, 2009, 10). Bei der Entwicklung von eigenen Modulen sollte sich an die Drupal-API gehalten werden. Drupal stellt Methoden und Hooks zur Verfügung, die bei dem Entwickeln neuer Module verwendet

werden können. Dies verhindert das Verändern des Drupal-Quellcodes, wodurch die Möglichkeit bestehen bleibt, ein Drupal-Update durchzuführen.

Die Benutzergruppen Au-pair und Gastfamilie haben die Möglichkeit, ein Profil in dem Portal anzulegen, um sich zu präsentieren. Das Anlegen des Profils und die Recherche nach geeigneten Au-pair bzw. Gastfamilien ist kostenlos. Außerdem können Au-pair den Gastfamilien (und umgekehrt) «Kontaktanfragen» senden. Das Bestätigen einer solchen Kontaktanfrage bescheinigt das gegenseitige Interesse. Um nun aber die Kontaktdaten dieses Benutzers sehen zu können, ist eine Premium-Mitgliedschaft nötig, über die sich das Portal auch finanziert. Bestreben des Portales ist es also, dass so viel Premium-Mitgliedschaften wie möglich abgeschlossen werden. Das Portal muss deshalb eine so hohe Usability aufweisen, dass es für einen Benutzer sehr einfach ist, passende Profile anderer Benutzer zu finden, um diese dann zu kontaktieren. Denn wenn ein Benutzer viele Kontaktanfragen hat, ist es sehr wahrscheinlich, dass er sich dazu entschließt, Premium-Mitglied zu werden. Nur so ist es für ihn möglich, die Kontaktdaten der Benutzer, die ihn kontaktiert haben, zu sehen.

Ein Ziel dieser Arbeit ist somit das Erhöhen der Usability zur Umsatzsteigerung. Auch ist ein Support für Benutzer, die das Portal nicht bedienen können, nahezu ausgeschlossen, da der Betreiber des Portals keine Angestellten hat. Ein weiteres Ziel ist demnach das Erhöhen der Usability zur Support-Minimierung.

Zur Steigerung der Usability im Portal sollte der Benutzer trotz weniger Einstellungen in einer Suchliste qualitativ gute Suchergebnisse erzielen können. Ein «Matchmaking» (Abschnitt 2.3 sowie 3.1) würde diese Anforderung erfüllen. Wenn der Benutzer dann durch eine Suchliste auf das Profil eines anderen Benutzers gelangt ist, wäre eine Funktionalität wünschenswert, die es ermöglicht, dem Benutzer andere Benutzerprofile, die Ähnlichkeiten mit dem gerade betrachteten Profil aufweisen, zu empfehlen. Diese Anforderung kann mit Hilfe von «Recommendations» (Abschnitt 2.4 sowie 3.2) umgesetzt werden.

## 2.3 Matchmaking

«Matchmaking» ist die „Vermittlung von Angebot und Nachfrage auf Märkten auf Basis von Profilinformatoren“ (Thiemann, 2006, 5). Matchmaking kann eine automatisierte Suche oder das automatische Verbinden und Verwenden geeigneter Services beinhalten (vgl. ebd.).

Benutzer, die sich in einem Online-Portal anmelden, geben im Registrierungsprozess eine Vielzahl persönlicher Informationen an. Warum sollte man diese Informationen nicht sofort nutzen? Man könnte diese Angaben bei einer automatisierten Suche, bei der einige der persönlichen Daten mit denen eines anderen Benutzers verglichen und «gematcht» werden, verwenden. Das heißt, das Portal-System nimmt ohne das Zutun des Benutzers eine oder mehrere der Informationen aus dem Profil, gleicht diese mit den anderen Benutzern ab und liefert nur zu diesem Profil passende Ergebnisse.

Ein Beispiel hierfür sind Partnerportale. In dem Online-Partner-Portal «eDarling» gehört es zum Registrierungsprozess, einen wissenschaftlichen Persönlichkeitstest durchzuführen. Somit werden sich ausschließlich Mitglieder vorgeschlagen, deren Persönlichkeit und soziodemographischer Hintergrund in einer Weise übereinstimmen, die eine langfristige und erfüllte Beziehung versprechen (vgl. eDarling, 2011).

Mit einer Suche, die dem Benutzer so wenig wie möglich selbst zur Eingabe auffordert, die aber durch ein passendes Matchmaking trotzdem qualitativ gute Ergebnisse liefert, kann man die Usability stark erhöhen. Steve Krug zeigt beispielsweise in seinem Buch «Don't make me think», wie man einen Benutzer in einer Suchliste mit nur einem einzigen Auswahlfeld verunsichern kann (vgl. dazu Krug, 2006, 16ff.). Er beschreibt den gedanklichen Ablauf eines Benutzers, welcher in einem Online-Buchhandel angeben kann, ob er den gesuchten Begriff in den Kriterien «Keyword», «Titel» oder «Autor» suchen möchte. Im Vergleich dazu nennt er den Versandhandelsriesen «Amazon.com», bei dem der Benutzer den Suchbegriff einfach in dem Kriterium «Bücher» suchen kann. Dabei stellt Steve Krug fest, dass es wichtig ist, Fragezeichen über den Köpfen der Benutzer zu eliminieren, indem man einen Benutzer nicht auffordert, unnötige Eingaben zu machen (vgl. ebd.). Mit Hilfe solcher Vereinfachungen erreicht man eine höhere Usability.



Um die Interaktion des Benutzers so gering wie möglich zu halten, empfiehlt sich die Verwendung eines Matchmakings. Man kann hiermit eine Fehlinterpretation des Suchfilters vorbeugen und den Benutzer bei der Suche nach dem passenden Angebot unterstützen. Beispiele wie «eDarling» zeigen, dass es verschiedene Ansätze für das Matchen von Benutzern gibt. Es ist dabei vollkommen irrelevant, ob das simple Verknüpfen der Profildaten der Benutzer oder ein psychologischer Test für das Matchen verantwortlich sind. Matchmaking bietet dem Benutzer immer eine zielgerichtete Variante einer Suchliste.

## 2.4 Recommendations

Das World Wide Web ist zum Massenmedium geworden. Der Umsatz mit Waren und Dienstleistungen, die über das Internet verkauft wurden, hat in den vergangenen Jahren stark zugenommen und wächst weiter (vgl. Runte, 2000, 23). „Da sich das Internet zunehmend als kommerzielle Plattform entwickelt und die erforderlichen Möglichkeiten zur Rationalisierung und zur computergestützten Interaktion bietet, ist es die wichtigste Plattform für Recommender Systeme“ (Schneider, 2005, 10).

Internet-Shops wie «Amazon» oder «ebay» setzen Recommendations – also Empfehlungen – ein, um ihren Umsatz zu optimieren. Es werden zu den gewählten Artikeln auch andere Artikel angezeigt, die sich laut ebay «andere Mitglieder im Anschluss an dieses Produkt angesehen haben» (vgl. ebay, 2011). Amazon hingegen zeigt Artikel, die oft mit dem gerade angezeigten Artikel zusammen gekauft wurden. Dies wird durch den folgenden Satz angezeigt: „Kunden, die diesen Artikel gekauft haben, kauften auch“ (amazon, 2011). Durch diese Anzeigen werden Benutzer des Online-Shops ohne weitere Interaktion auf Artikel aufmerksam gemacht, auf die sie durch ihre eigentliche Suche nicht gekommen wären. Empfehlungen können demnach den Umsatz steigern, da dem Benutzer aufgrund seiner vorherigen Auswahl zielgerichtet ähnliche Produkte angeboten werden.

Mit Empfehlungen kann man auch Benutzern eine Möglichkeit schaffen, andere interessante Benutzerprofile zu finden. Beispielsweise werden auf dem Jugend-Online-

Portal von Chemnitz <http://www.triff-chemnitz.de> die besten zehn weiblichen und männlichen Profile angezeigt. Die Seite betreibt einen «HotVote», bei dem es für Benutzer möglich ist, andere Benutzerprofile auf einer Skala von eins (eher schlecht) bis zehn (hot) zu bewerten. Die besten Zehn werden dann in der HotVote-Liste angezeigt. Diese Liste ist auch von nicht-angemeldeten Benutzern zu erreichen. Jemand, der also mit diesen zehn «besten Benutzern» in Kontakt treten möchte, meldet sich kostenlos im Jugend-Online-Portal an. Dies entspricht dem Hauptziel des Portales. Denn diese Seite ist auf eine hohe Benutzeranzahl angewiesen, da sie sich ausschließlich mit Werbeeinblendungen finanziert. Die HotVote-Liste repräsentiert im weitesten Sinne die Bestseller<sup>5</sup> eines Online-Shops.

«Amazon» und «ebay» setzen Recommendations ein, um Produkte anzuzeigen, auf die der Benutzer eventuell nicht aufmerksam geworden wäre. Die beiden Online-Versandhändler versuchen dabei, immer einen Bezug zu dem Produkt herzustellen, wofür sich der Benutzer gerade interessiert. Die Chance, so weitere Produkte zu verkaufen, steigt dadurch erheblich. Dass man aber nicht ausschließlich Produkte empfehlen muss, zeigt das Beispiel von <http://www.triff-chemnitz.de>. Es ist also eine sinnvolle Variante, dem Benutzer Recommendations anzuzeigen, um den Verkauf zu steigern. Durch die weite Verbreitung von «Amazon» und «ebay» und deren Einsatz von Recommendations ist der Umgang mit diesen für den Benutzer vertraut, sodass man mit dem Einsatz auch die Usability steigern kann. Beides sind Ziele des hier behandelten Online-Portals, was für den Einsatz von Recommendations spricht.

## 2.5 A/B-Test

Eine Art, den Erfolg oder Misserfolg einer Website zu messen, ist ein A/B-Test. Üblicherweise misst man im Internet mit einem A/B-Test die «conversion rate»<sup>6</sup> einer Werbeeinblendung. Man kann dadurch analysieren, ob die Benutzer eher durch

---

<sup>5</sup> Als Bestseller werden Produkte bezeichnet, die innerhalb einer bestimmten Periode besser als vergleichbare Produkte verkauft worden sind.

<sup>6</sup> Ist der Anteil der Benutzer, die eine gewünschte Aktion durchführen (vgl. [marketingterms.com](http://marketingterms.com), 2011). Bei einem Banner wäre dies der Klick auf den Banner.

den ersten oder zweiten Banner auf die eigene Website gekommen sind und bei welchen Benutzern dann dieser Besuch zu einem Kauf führte. Zum Schluss kann man den Erfolg jeder Werbeeinblendung an den daraus resultierenden Käufen messen.

Geht man zusätzlich davon aus, dass eine Website nur dann Erfolg aufweist, wenn die Usability dieser sehr gut ist, so kann man mit einem A/B-Test auch die Usability einer Website messen.

Bei einem A/B-Test erstellt man zwei Varianten einer Website. Beide haben aber das selbe Ziel, beispielsweise den Verkauf eines Produktes. Die Gesamtheit der Benutzer wird per Zufall in zwei Gruppen – beispielsweise 50 % zu 50 % – eingeteilt. Diese beiden Gruppen repräsentieren dann die Gruppen A und B. Die Gruppe A wird der ersten und die Gruppe B der zweiten Variante der Website zugewiesen. Wenn das Produkt überwiegend von Gruppe B gekauft wird, so scheint die Variante B – und damit auch die Usability – dieser Website besser zu sein.

Dass es bei einem A/B-Test nicht nur um den Kauf einer Ware gehen kann, zeigt das Beispiel von Dustin Curtis. Er wollte auf seiner Website Benutzer davon überzeugen, dass sie ihm auf «Twitter»<sup>7</sup> folgen. Mittels eines A/B-Testes konnte er feststellen, dass durch geschicktes Ändern des Link-Textes die CTR von 4,70 % auf 12,81 % gesteigert werden konnte (vgl. Curtis, 2011). Dieses Experiment zeigt auf einfache, aber sehr eindrucksvolle Weise, wie ein A/B-Test systematisch ablaufen kann. Daher wird im Folgenden näher auf diesen Test eingegangen.

Das Ziel, welches Dustin Curtis mit seiner Website erreichen wollte, war das Finden neuer «follower»<sup>8</sup> bei Twitter. Er benutzte dafür einen simplen Text-Link: „I’m on Twitter.“ (ebd.). Dieser Text war Ausgangspunkt seines A/B-Testes. Über diesen Link generierte er 4,70 % CTR. Im ersten Schritt wechselte er von dieser indirekten zu einer direkten Ansage an den Benutzer: „Follow me on twitter.“ (ebd.). Diese Art der Ansprache an den Benutzer führte zu einem Sprung um 55 % auf eine CTR

---

<sup>7</sup>Twitter ist ein Informationsnetzwerk, welches in Echtzeit die neuesten Informationen zu einem Thema liefert, die den Benutzer interessieren. Das Interesse an einem solchen Thema bekundet man, indem man einem Benutzer oder Thema «folgt» (vgl. Twitter, 2009).

<sup>8</sup>Follower sind die Benutzer, die Interesse an dem Thema haben und sich weiterhin darüber informieren wollen.

von 7,31 %. Mit dem nächsten Link-Text erhöhte er den Druck auf die Besucher seiner Website nochmals: „You should follow me on twitter.“ (Curtis, 2011). Mit dieser veränderten Anrede erreichte er eine CTR von 10.09 %, was eine Steigerung um weitere 38 % zum vorhergehenden Text bedeutete. Mit seinem letzten Link-Text zeigte er dem Benutzer sogar noch, *wie* der Benutzer ihm am besten folgen kann: „You should follow me on twitter here.“ (ebd.). Dieser Text verhalf ihm mit einer Steigerung von 27 % auf die abschließende CTR von 12,81 %. Insgesamt konnte Dustin Curtis seine CTR vom Ausgangspunkt bis zu seiner letzten Variante demnach um 173 % steigern (vgl. dazu ebd.).

Ein strukturiert systematisch ablaufender A/B-Test mit einem definierten Ziel kann einem Betreiber einer Website sehr viele Informationen über die Vorlieben der Benutzer geben, mit denen er nicht gerechnet hätte. Die Website «CareLogger.com» erhöhte beispielsweise die «conversion rate» um 34 % durch das simple Ändern der Farbe des Anmelde-Buttons von Grün auf Rot (vgl. Chopra, 2010).

Ein Vorteil eines A/B-Testes ist es also, die «conversion rate» einer Website und somit eventuell den Umsatz dieser Website zu steigern. Für das Erstellen von A/B-Tests auf einer Website gibt es im Internet zahlreiche Tools. Das bekannteste Tool – der «Google Website Optimizer» – ist sogar kostenlos. Es unterstützt zum einen beim Erstellen des Testes und zum anderen beim Auswerten der Testdaten.

Abschließend ist festzuhalten, dass mit einem A/B-Test kostenlos der Umsatz der eigenen Website gesteigert werden kann. Diese positiven Aspekte sollte jeder Betreiber einer Website nutzen.

## 3 Implementierung

### 3.1 Matchmaking

Wie bereits erwähnt, ist das Matchmaking eine einfache Möglichkeit, einem Benutzer eine nach seinen persönlichen Bedürfnissen angepasste Suchliste anzubieten. Die Suchlisten im Portal werden für die Suche nach Au-pair / Gastfamilien verwendet. Dabei stehen dem Benutzer Filter zur Verfügung, die den Angaben der Benutzer, die sie schon bei der Registrierung gemacht haben, entsprechen. Die vorhandenen Informationen der registrierten Benutzer können miteinander verknüpft, das heißt «gematcht» werden.

#### 3.1.1 Konzept

Bei der Implementierung des Matchmakings soll darauf geachtet werden, dass es sich in die vorhandene Struktur einfügt.

Um den Bestandteil «Fehlertoleranz» der Usability-Attribute zu erfüllen, muss die generierte Suche trotz fehlender oder fehlerhafter Elemente ein Ergebnis liefern, mit welchem der Benutzer dennoch arbeiten kann. Da der Benutzer im Umgang mit dem Portal schon mit anderen Suchlisten – wie bei der einfachen Suche als Gast oder bei der erweiterten Suche als registriertes Mitglied – gearbeitet hat, ist er mit diesen bereits vertraut und sollte somit keine Probleme haben, eine neue Suchliste zu bedienen. Dies entspricht den Usability-Attributen der einfachen «Erlernbarkeit» und «Merkbarkeit».

Wie angesprochen, muss ein Benutzer, der sehr detailliert nach seinen Vorstellungen

suchen möchte, eine Vielzahl von Filtern mit Werten versehen, die er eigentlich bereits bei seiner Registrierung angegeben hat. Da die Filtereingaben außerdem nicht gespeichert werden, ist der Benutzer dazu gezwungen, immer wieder die selben Angaben über sich zu machen. Dies ist sehr zeitaufwendig und frustriert vermutlich den Benutzer. Um dies zu verhindern, können bei einer Suche die Benutzerinformationen automatisch gematcht werden. Damit wird die «Effizienz» – ein weiteres Usability-Attribut – sehr gesteigert. Geht man davon aus, dass ein Benutzer beim Anwenden einer effizienteren Suche schneller an – für ihn – relevantere Ergebnisse kommt, erfüllt man auch das fünfte Usability-Attribut nach Nielsen: «die Zufriedenheit».

Um Felder zu definieren, die für eine Matchmaking-Suche am besten geeignet sind, muss der Registrierungsprozess der jeweiligen Benutzergruppen im Portal untersucht werden. Um beispielsweise kein Feld abzufragen, welches vom Benutzer nicht gesetzt sein könnte, empfiehlt es sich, auf Felder zurückzugreifen, die bei der Registrierung zwingend angegeben werden müssen. Auf der untersuchten Portal-Seite hat jeder Benutzer – ob dort angemeldet oder nicht – die Möglichkeit, auf einer einfachen Suchseite nach passenden Au-pair / Gastfamilien zu suchen. Bei dieser Suche ist es für die Gastfamilien lediglich möglich, nach dem Geschlecht der Au-pair und deren Heimatland zu filtern. Darüber hinaus kann die Gastfamilie noch angeben, aus welchem Land sie selbst kommen. Die Au-pair können in dieser einfachen Suche die Liste der Gastfamilien mit der Angabe ihres Geschlechtes, ihres Heimatlandes und des Heimatlandes, aus dem die Familie kommen soll, filtern.

Ist ein Benutzer im Portal angemeldet, kann davon ausgegangen werden, dass er diese einfache Suche – die ihm, wie beschrieben, die ganze Zeit zur Verfügung stand – verfeinern möchte. Hier wäre also ein Ansatz, diese Filter bereits zu setzen und andere zusätzliche Filter anzubieten, sodass der Benutzer seine Suche vertiefen und effizienter nutzen kann.

Eine weitere Anforderung an die Suche besteht darin, die Werte aus dem Feld «gute Sprachkenntnisse» in das Matchmaking aufzunehmen, da es sehr wichtig ist, sich mit dem Au-pair / der Gastfamilie verständigen zu können.

Für die Matchmaking-Suche wurde sich somit auf die vier folgenden Au-pair-Felder geeinigt:

- `field_gender`: In diesem Feld ist die Angabe des Geschlechtes des Au-pair gespeichert.
- `field_home`: Dieses Feld beinhaltet das Heimatland des Au-pair.
- `field_go_to_land`: In diesem Feld stehen die Länder, in denen das Au-pair gern arbeiten möchte.
- `field_good_lang`: Dieses Feld gibt die guten Sprachkenntnisse des Au-pair an.

Im Vergleich dazu wurden die vier folgenden Felder der Gastfamilie verwendet:

- `field_gender_search`: Dieses Feld beinhaltet das Geschlecht, welches das Au-pair haben soll.
- `field_from`: Hier wird gespeichert, aus welchem Heimatland das Au-pair sein soll.
- `field_home`: In diesem Feld gibt die Familie ihr Heimatland an.
- `field_good_lang`: Dieses Feld gibt die guten Sprachkenntnisse der Gastfamilie an.

Da diese Felder in der Datenbank gespeichert werden, kann man mit einem geeigneten SQL-Statement die Verbindungen unter den einzelnen Feldern feststellen. Ein erster Ansatz zur Realisierung einer Matchmaking-Suche war es, mittels SQL-Statement die Verbindung der Profile herzustellen, um diese in einer Liste anzugeben. Im Anschluss daran sollte der Benutzer diese Liste mit weiteren Filtermöglichkeiten eingrenzen können. Bei der Umsetzung dieses Ansatzes ergaben sich nachstehende Probleme: Erstens wurde das SQL-Statement für das Matchen im Quelltext angegeben. Damit wurde die Flexibilität des Matchmakings stark eingegrenzt. Zweitens entstand erheblicher Aufwand, um die Ergebnisse dieses SQL-Statements als Suchliste so

in das Portal-System einzugliedern, dass sich diese Suchliste nicht von den bereits vorhandenen abgrenzt. Auf diese Weise eine Matchmaking-Suche zu realisieren, wurde wegen der genannten Probleme verworfen.

Die bereits bestehenden Suchlisten wurden im Portal-System mit dem Drupal-Modul «Views» erstellt. Dies ist ein Modul, welches mit genug Informationen ein SQL-Statement erstellen, ausführen und dessen Ergebnisse anzeigen kann (vgl. Views, 2011). Aufgrund der Einstellungsmaske von Views sind beim Erstellen einer View keine SQL-Kenntnisse notwendig. Außerdem sind keine PHP-Kenntnisse für die Formatierung der Ausgabe nötig. Anpassungen können aber trotzdem mit der Einstellungsmaske realisiert werden. Aufgrund der bestehenden Suchlisten und der Vorteile des Moduls Views entschied man sich für den Ansatz, mit Views die Matchmaking-Suche zu realisieren.

Um diese Funktionalität zu gewährleisten, wurde das Modul «matchmaking» entwickelt. Dieses Modul ist kein eigenständiges Modul, sondern eher ein «Plug-in»<sup>1</sup> für Views, welches die Felder, die gematcht werden sollen, ausliest, vergleicht und in das SQL-Statement von Views einfließen lässt.

Bei dem Vergleich der Felder gibt es eine portalspezifische Eigenheit, auf die geachtet werden muss: Es ist nötig, bestimmte Werte dem SQL-Statement zu übergeben und andere Werte wiederum aus diesem auszuschließen. Ein Beispiel dafür ist das Matchen des Geschlechtes. Au-pair können im Portal ihr Geschlecht – männlich oder weiblich – angeben. Die Gastfamilien haben bei der Wahl des Geschlechtes ihres Au-pair aber noch eine zusätzliche Option: Sie können hier auch «egal» angeben. Bei dem Matchen des Geschlechtes der Au-pair und des gesuchten Geschlechtes der Gastfamilie muss diese Eingabe der Gastfamilien berücksichtigt werden, sodass bei einer Suche der Au-pair alle Gastfamilien angezeigt werden, die zum einen das passende Geschlecht und zum anderen «egal» angegeben haben.

Um die Entwicklung des Moduls «matchmaking» zu erläutern, muss man zunächst auf die Funktionalität von Views zusammen mit dem Modul «Content Construction Kit» (kurz CCK) eingehen, da die Felder für das Matchen von CCK zur Verfügung gestellt und in Views integriert werden.

---

<sup>1</sup>Ist eine Erweiterung einer bestehenden Software.



*Exkurs: CCK-Integration in Views*

Das Modul CCK erlaubt es, benutzerdefinierte Felder auf Inhaltsseiten – den «Nodes» – anzuzeigen. Diese Felder enthalten die persönlichen Informationen, die ein Benutzer bei der Registrierung im Online-Portal angeben muss. Mit Hilfe dieser Felder werden auch die bereits vorhandenen Suchlisten erstellt. Views kann auf die CCK-Felder zugreifen und deren Wert auslesen, sodass auf einer Suchliste beispielsweise nur nach weiblichen Au-pair gefiltert werden kann. Beim Definieren der CCK-Felder ist es möglich, einen Feldtyp anzugeben. Dieser Feldtyp bestimmt den «Views-Handler», der auf dieses Feld angewendet wird. Die Abbildung 3.1 zeigt die «Filter-Handler», die beim Erstellen einer View zur Auswahl stehen, wenn ein Filter angelegt wird:

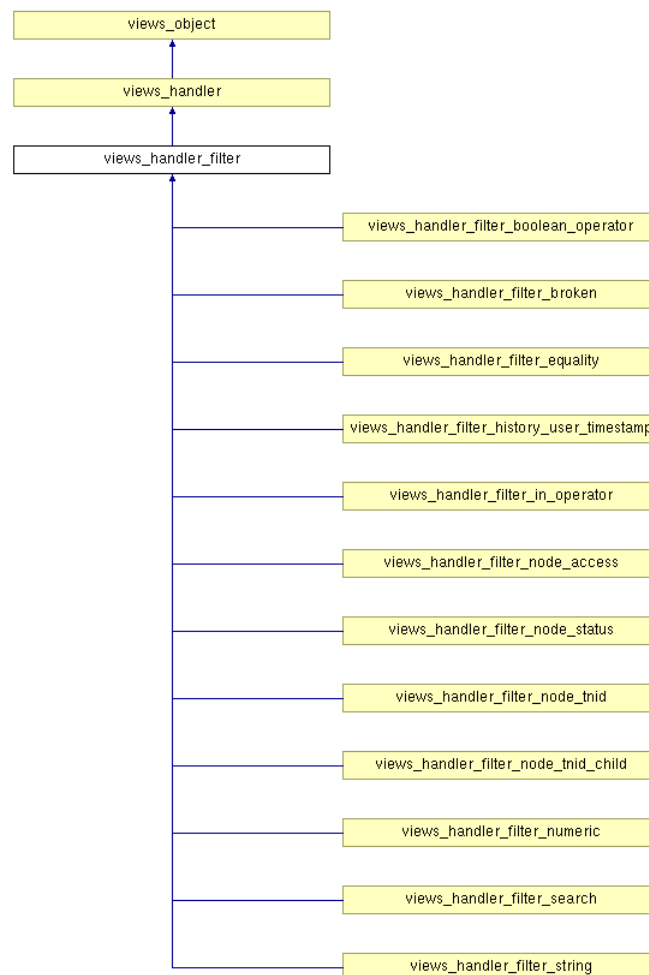


Abbildung 3.1: Von Views bereitgestellte Filter-Handler

Ist ein Feldtyp beispielsweise «Ganzzahl», so wird der Filter-Handler «views\_handler\_filter\_numeric» verwendet. Dieser Handler gibt nun die Möglichkeit, numerisch spezifische Angaben anzugeben, wie zum Beispiel einen Vergleich mit «größer als» oder «kleiner als». Diese Auswahl würde bei einem Textfeld keine gültigen Resultate liefern. Darum wird bei einem Feld, das den Typ «Text» hat, der Filter-Handler «views\_handler\_filter\_string» verwendet. Dieser ermöglicht es unter anderem, nach einem definierten String zu suchen und lässt Operationen wie «größer als» oder «kleiner als» außer Acht.

Es ist nun klar, wie Views auf die CCK-Felder reagiert und diese verwendet. Das Matchmaking-Modul nutzt diesen beschriebenen Ansatz.

Views stellt demzufolge mit den Filtern und deren Handlern bereits einen Großteil der Funktionalität eines Matchmakings zur Verfügung. Die Filter können auf alle Felder angewendet werden und sind über die Einstellungsmaske von Views sehr leicht konfigurier- und austauschbar. Views soll also mit einem geeigneten Modul um die fehlende Funktionalität – dem Vergleichen von CCK-Feldern mit den Feldern des angemeldeten Benutzers – erweitert werden. Auf die genaue Implementierung wird im nächsten Abschnitt eingegangen.

### 3.1.2 Implementierung

Um das Matchmaking-Modul in Views zu verwenden, muss das Modul lediglich an die Views-API angebunden werden. Der folgende Quelltext realisiert dies:

Listing 3.1: Matchmaking-Anbindung an die Views-API

```
1 function matchmaking_views_api() {  
2     return array(  
3         'api' => 2,  
4         'path' => drupal_get_path('module', 'matchmaking'),  
5     );  
6 }
```

Mit dem Hook «matchmaking\_views\_api» registriert man zunächst die View-API-Informationen. Dies ist notwendig, um weitere dem Modul angehörende Dateien zu laden (Drupalcontrib, 2011b). Die Angabe der API-Version ist Pflicht. Im Portal wird

derzeit die Views-Version 2.12 eingesetzt. Es handelt sich also um die API-Version 2, welche mit dem Parameter «api» angegeben wird. Der Parameter «path» teilt der Views-API mit, wo die dem Modul angehörenden Dateien zu finden sind.

### 3.1.2.1 Views-Filter-Handler

Wie in der Abbildung 3.1 zu sehen ist, existiert ein Filter-Handler «views\_handler\_filter\_equality». Dieser ist eine Oberklasse von Filtern und wird von anderen Filtern verwendet, um Werte auf «gleich» oder «ungleich» zu überprüfen. Diese Funktionalität wäre für ein Matchmaking passend gewesen. Leider ist der zu überprüfende Wert statisch anzugeben. Die Funktionalität, diesen Wert aus den Profilen der Benutzer zu laden und mit einem definierten Feld zu vergleichen, fehlt. Da es ebenfalls nicht vorgesehen ist, Filter-Handler zu definieren und diese dann einem bestimmten Feldtyp zuzuweisen, musste ein neuer Filter-Handler erstellt werden. Der folgende Quelltext beschreibt die Definition eines neuen Handlers, damit Views diesen Handler findet:

Listing 3.2: Definition eines neuen Filter-Handlers

```
1 function matchmaking_views_handlers() {  
2     return array(  
3         'info' => array(  
4             'path' => drupal_get_path('module', 'matchmaking'),  
5         ),  
6         'handlers' => array(  
7             'matchmaking_handler_filter_match' => array(  
8                 'parent' => 'content_handler_filter_string',  
9             ),  
10        ),  
11    );  
12 }
```

Man kann mit dem Hook «matchmaking\_views\_handler» in das Erstellen der Filter-Handler eigene Funktionalität integrieren. Die beiden Array-Felder, die dafür angegeben werden müssen, sind «info» (Zeile 3) und «handlers» (Zeile 6). «info» beinhaltet die Informationen darüber, wo die zu dem Modul gehörigen Dateien

sind. In dem «handlers»-Array wird der Name des neuen Handlers definiert. Zeile 7 des Quelltextes definiert also einen neuen Filter-Handler namens «matchmaking\_handler\_filter\_match». Dies ist wiederum ein Array, in dem Felder angegeben werden können. In diesem Array wurde – wie in Zeile 8 zu sehen ist – das Array-Feld «parent» übergeben. Der neue Filter-Handler erbt dadurch von dem bereits vorhandenen Filter-Handler. Somit wurden sämtliche Funktionalitäten der Filter-Handler beibehalten und zusätzliche Funktionen durch den neuen Handler angehängen. Obwohl Views ein neuer Filter-Handler hinzugefügt wurde, kann dieser noch nicht verwendet werden. Dies wird mit einem Hook erreicht. Im folgenden Quelltext wird auf diese Realisierung genauer betrachtet:

Listing 3.3: Views-Hook für die Filter-Handler-Verwendung

```

1 function matchmaking_views_data_alter(&$data) {
2     $fields = content_fields();
3     foreach ($fields as $fieldkey => $value) {
4         $tablename = content_views_tablename($value);
5         $dbinfo = content_database_info($value);
6         $columnname = $dbinfo['columns'][$value]['column'];
7         if (isset($data[$tablename][$columnname]['filter']['handler']))
8             {
9                 $data[$tablename][$columnname]['filter']['handler'] = '
10                 matchmaking_handler_filter_match';
11             }
12     }
13 }
```

In der Funktion «views\_data»<sup>2</sup> wird die Tabellenstruktur von Views beschrieben. In diese Funktion wird sich mittels des hier verwendeten Hooks «matchmaking\_views\_data\_alter» angehängen, um zusätzliche Tabellenstruktur zu definieren. In Zeile 2 werden mit der Methode «content\_fields» die CCK-Feldinformationen geladen und in der Variable «\$fields» gespeichert. Die «foreach» aus Zeile 3 liest nun alle Felder aus der Datenbank und speichert dabei Tabellennamen, Spaltennamen und die Datenbankinformationen jedes Feldes. Dieses Feld wird mit der if-Verzweigung in Zeile 7 dahingehend überprüft, ob ein Filter-Handler gesetzt ist. Ist dies der Fall, wird der definierte Filter-Handler durch den eigenen überschrieben. Alle CCK-Felder

<sup>2</sup>Ist eine Standard-Drupal-Funktion und wird hier nicht angezeigt.

bekommen so den Filter-Handler «matchmaking\_handler\_filter\_match». Damit sind die grundlegenden Voraussetzungen für das Verwenden eines eigenen Filter-Handlers erfüllt. Nun wird auf die Funktionalität des Filter-Handlers eingegangen.

Die Konfiguration des neuen Filters soll äquivalent zu den bestehenden Filtern sein. Daher müssen zum einen ein neuer Operator, welcher die Einstellungsmaske aufrufen kann, und zum anderen die Einstellungsmaske selbst zur Verfügung stehen.

Zunächst wird auf das Bereitstellen des Operators eingegangen:

Listing 3.4: Definition eines neuen Filter-Operators

```
1 public function operators() {  
2     $operators = parent::operators();  
3     $operators['cck'] = array(  
4         'title' => t('Is equal to CCK'),  
5         'short' => t('cck'),  
6         'method' => 'op_cck',  
7         'values' => 2,  
8     );  
9     return $operators;  
10 }
```

Auch hier werden die vorhandenen Operatoren durch Vererbung beibehalten und durch den neuen Operator ergänzt. Zeile 2 verdeutlicht dies. In Zeile 3 wird der Variable «\$operators» ein neuer Wert – und somit ein neuer Operator – «CCK» hinzugefügt. Das Array, welches hier geöffnet wird, enthält alle Einstellungen zu diesem Operator. Bei der Verwendung in Views wird der Titel «Is equal to CCK» eingesetzt (Zeile 4). Das Feld «short» wurde auf «cck» gesetzt. Die Methode, die bei dem Operator ausgeführt werden soll, ist die «op\_cck»-Methode. Im folgenden Abschnitt wird diese Methode näher betrachtet, da sie die Funktionalität des Moduls beinhaltet. In dieser Methode wird unter anderem das SQL-Statement für die Matchmaking-Suche bereitgestellt<sup>3</sup>:

---

<sup>3</sup>Der Quelltext wurde in drei Teile gegliedert, um ihn besser erläutern zu können.

Listing 3.5: op\_cck-Methode Teil 1: Speichern von Benutzerdaten in das \$value\_list-Array

```

1 function op_cck($field , $upper) {
2     $scan_be_null = false;
3     $value_list = array();
4     if (is_array($this->value)) {
5         foreach ($this->value as $value) {
6             if (!empty($value['value'])) {
7                 $value_list[] = $value['value'];
8             }
9         }
10    }

```

Zunächst wird in der «op\_cck»-Methode die boolsche Variable «\$scan\_be\_null» auf «false» gesetzt (Zeile 2) und ein leeres Array «\$value\_list» initialisiert (Zeile 3). Die CCK-Felder, die einer View übergeben werden, sind in einem Array gespeichert. Zuerst wird in Zeile 4 überprüft, ob es sich um ein Array handelt. Wenn es ein Array ist, wird dieses in Zeile 5 mit einer foreach-Schleife durchlaufen.

Da CCK alle Felder in einem Array übergibt und in dem neuen Array «\$value\_list» aber keine Felder ohne Wert stehen sollen, muss mit der if-Verzweigung aus Zeile 6 darauf überprüft werden, ob in dem Feld überhaupt ein Wert gespeichert ist. Ist dies der Fall, dann wird der Wert dieses Feldes in das Array «\$value\_list» gespeichert. Diese Werte werden später in das SQL-Statement eingefügt.

Bevor nun die Werte im SQL-Statement ergänzt werden, muss noch auf die Werte eingegangen werden, die eventuell zusätzlich hinzugefügt werden müssen (vgl. dazu Abschnitt 3.1.1). Der folgende Quelltextabschnitt erläutert den Umgang mit der «ignore\_value». Diese Werte sollen entweder zusätzlich oder gar nicht in das SQL-Statement. Mit Hilfe der Konfigurationsmaske dieses Moduls kann man angeben, ob es einen solchen Wert gibt und ob dieser einfließen soll oder nicht.

Listing 3.6: op\_cck-Methode Teil 2: Speichern der \$ignore\_value in das \$value\_list-Array

```

1 if (in_array($this->cck_ignore_value , $value_list)) {
2     return;
3 }

```

```

4      if ($this->cck_ignore_include == 1) {
5      if ($this->cck_ignore_value == '0' || !empty($this->
        cck_ignore_value)) {
6          $value_list[] = $this->cck_ignore_value;
7      } else {
8          $scan_be_null = true;
9      }
10 }

```

In Zeile 1 wird geprüft, ob die definierte «ignore\_value» schon mit dem im Profil angegebenen Wert übereinstimmt. Sollte dies der Fall sein, muss überhaupt keine Filterung erfolgen, denn die «ignore\_value» beinhaltet meist Werte wie «egal» oder «alle». Benutzer, die diese Werte angegeben haben, wollen an dieser Stelle offensichtlich keine Filterung der Ergebnisse.

Zeile 4 überprüft mit der if-Verzweigung, ob in der Einstellungsmaske die Angabe gemacht wurde, dass die «ignore\_value» mit in das SQL-Statement einfließen soll. SQL kann nicht nur auf bestimmte Werte überprüfen, sondern auch, ob überhaupt ein Wert existiert. Hierbei ändert sich aber der Quelltext im SQL-Statement. Daher muss in Zeile 5 auf den Wert dieser «ignore\_value» überprüft werden. Ist der Wert 0, so wird dieser auch in das Array gespeichert. Ist keiner angegeben, wird die Variable «\$scan\_be\_null» gesetzt. Diese Variable dient dazu, den Quelltext des SQL-Statements anzupassen.

Der folgende Quelltext-Abschnitt beschreibt, wie die zu matchenden Werte dem SQL-Statement hinzugefügt werden:

Listing 3.7: op\_cck-Methode Teil 3: SQL-Statement mit Werten des \$value\_list-Arrays erzeugen

```

1 $sql_string = "$upper($field) IN $upper(";
2 $first = true;
3 foreach ($value_list as $value) {
4     if ($first) {
5         $first = false;
6     } else {
7         $sql_string .= ',';
8     }
9     $sql_string .= "'%s'";
10 }

```

```

11     $sql_string .= ")";
12     if ($can_be_null) {
13         $sql_string .= " OR $upper($field) IS NULL";
14     }
15     if (!empty($value_list)) {
16         $this->query->add_where($this->options['group'],
17             $sql_string, $value_list);
18     }

```

In der ersten Zeile wird eine Variable «\$sql\_string» initialisiert und speichert den ersten Teil des SQL-Scriptes. Die SQL-Funktion «upper» bewirkt, dass alle Feldnamen in Großbuchstaben übergeben werden. Die Variable «\$sql\_string» wird schrittweise mit der für das Matchen notwendigen SQL-Syntax erweitert. Zeile 3 durchläuft zunächst das gesamte «value\_list»-Array und fügt der Variable «\$sql\_string» für jedes Element in diesem Array ein «'%s'» hinzu (Zeile 9). Da diese Werte mit Komma getrennt werden müssen, wird mit der if-Verzweigung aus Zeile 3 untersucht, ob es sich bei diesem Hinzufügen um das erste Element handelt oder nicht. Ist es nicht das erste Element, wird im else-Zweig auf Zeile 7 ein «,» an den Wert der Variable «\$sql\_string» angehängt. Zeile 11 schließt die SQL-Syntax mit einer schließenden Klammer «)» ab. Die Variable besitzt zu dieser Zeit folgenden Wert<sup>4</sup>:

Listing 3.8: Beispiel-Wert für die Variable \$sql\_string

```

1 $upper($field) IN $upper('%s')

```

In Zeile 12 wird mit einer if-Verzweigung der Status der Variable «\$can\_be\_null» überprüft. Sollte diese TRUE sein, fließt eine leere «ignore\_value» in das SQL-Statement ein. Wenn dieser Fall eintritt, wird in Zeile 13 ein «OR \$upper(\$field) IS NULL» als Schluss an die SQL-Syntax der Variable «\$sql\_string» hinzugefügt. Die folgende Quelltextzeile verdeutlicht, welchen Wert die Variable «\$sql\_string» dann hat:

Listing 3.9: Beispiel-Wert für die Variable \$sql\_string nach Überprüfung der «ignore\_value»

```

1 $upper($field) IN $upper('%s') OR $upper($field) IS NULL

```

<sup>4</sup>Zur Verdeutlichung wird davon ausgegangen, dass *ein* Wert gematcht werden soll.



Zeile 15 überprüft zur Sicherheit, ob das «value\_list»-Array nicht leer ist. Wenn dies nicht der Fall und das Array also wie beim Administrator leer ist, wird die SQL-Syntax nicht in das SQL-Statement eingefügt. Das leere «value\_list»-Array zeigt an, dass der Benutzer kein ContentProfile besitzt. Es gibt also keine zu matchenden Daten und das SQL-Statement würde fehlerhaft abbrechen.

Ist das Array aber nicht leer, so wird in Zeile 16 das SQL-Statement um eine WHERE-Klausel erweitert. Diese WHERE-Klausel wird wiederum mit der SQL-Syntax aus der «\$sql\_string»-Variable gefüllt. Der erste Parameter «\$this->options['group']» gibt an, ob die Felder mit AND oder OR verknüpft werden sollen. Da diese GROUP noch nicht existiert, werden diese Felder mit AND verknüpft (vgl. Drupalcontrib, 2011c). Der zweite Parameter übergibt die SQL-Syntax. Die Syntax beinhaltet derzeit für jedes zu matchende Feld den Platzhalter «%s» (Zeile 9), da es sich um einen String handelt. Welche Werte letztendlich für dieses «%s» eingesetzt werden sollen, steht im dritten Parameter der Funktion, dem «value\_list»-Array.

### 3.1.2.2 Matchmaking-Einstellungsmaske

Bei der Entwicklung des Moduls wurde eine Einstellungsmaske erstellt, auf die in diesem Abschnitt eingegangen werden soll. Die Maske wird mit Hilfe von Bildern beschrieben, damit die Funktionen besser verdeutlicht werden können, als es mit dem Beschreiben des Quelltextes möglich wäre. Da es sich um ein Plug-in<sup>5</sup> für Views und einen überschriebenen Filter-Handler handelt, muss man in einer View einen Filter hinzufügen, um das Matchen zu nutzen. Abbildung 3.2 zeigt das von Drupal bereitgestellte Fenster zum Erstellen von neuen Filtern:



Abbildung 3.2: Views-Einstellungsmaske: Filter hinzufügen

<sup>5</sup>Mit Plug-ins lassen sich Software-Produkte mit zusätzlichen Funktionen erweitern.

Danach muss das Feld angegeben werden, welches für den Filter verwendet wird (Abbildung 3.3):



Abbildung 3.3: Views-Einstellungsmaske: Feld hinzufügen

Um anzugeben, wie der Filter auf dieses Feld wirken soll, muss man einen Operator angeben. Hier ist der durch das Matchmaking-Modul bereitgestellte Operator «Is equal to CCK» zu verwenden (Abbildung 3.4):

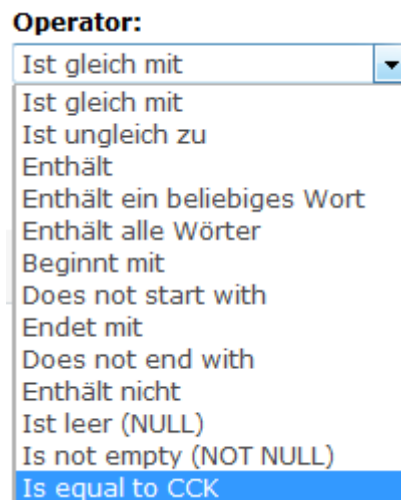


Abbildung 3.4: Views-Einstellungsmaske: Operator hinzufügen

Ist dieser Operator ausgewählt, wird man darauffolgend aufgefordert, ein Feld anzugeben, mit dem das Filter-Feld verglichen werden soll.

In diesem Beispiel wird also das Feld «field\_gender» des Au-pair mit dem Feld «field\_gender\_search» der Gastfamilien verglichen. Abbildung 3.5 veranschaulicht dies:

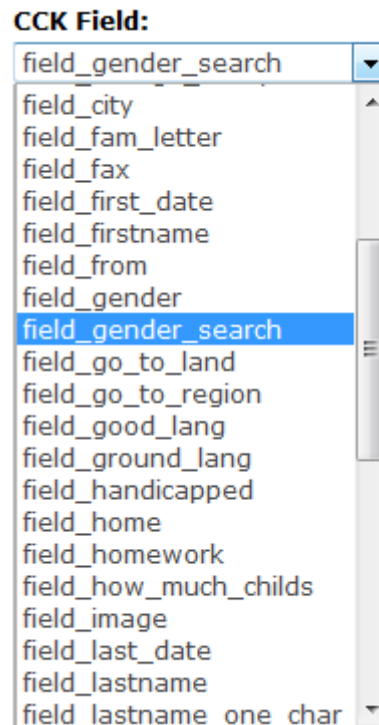


Abbildung 3.5: Views-Einstellungsmaske: Referenz-Feld hinzufügen

Die Gastfamilien haben – wie bereits erwähnt – die Möglichkeit, bei der Suche des Au-pair in dem Feld «Geschlecht» die Angabe «egal» zu machen. Sollte dieser Wert gesetzt sein, kann dieses Filter-Feld ignoriert werden; die «ignored\_value» wird also auf den Key des Wertes «egal» gesetzt. Die Angabe «egal» hat den Key «0». Die Abbildung 3.6 stellt die komplette Einstellungsmaske dar:

**Au-pair-Suche: Filter Inhalt: Geschlecht (field\_gender) konfigurieren**

☒ Status: Übersteuerte Werte werden verwendet.
 Standard verwenden

Der Eintrag ist derzeit nicht hervorgehoben. Sollte dieser **hervorgehoben** werden, können Benutzer den Filter ändern.
 Hervorheben

**Operator:** Is equal to CCK **CCK Field:** field\_gender\_search

**Ignored CCK Field Value:** 0
   
☐ Include ignored value in query

☒ Groß- und Kleinschreibung berücksichtigen
   
Filter die Groß-/Kleinschreibung beachten, können schneller sein. MySQL könnte die Groß-/Kleinschreibung ignorieren.

Aktualisieren Abbrechen Entfernen

Abbildung 3.6: Matchmaking-Einstellungsmaske

Als Folge dieses Filters werden einer Gastfamilie alle Au-pair angezeigt, die das gesuchte Geschlecht haben. Hat die Gastfamilie «egal» angegeben, wird kein Filter verwendet. Somit werden alle Au-pair angezeigt.

### 3.1.3 Fazit

Abschließend ist zu sagen, dass ein Modul entwickelt wurde, welches ein Matchen verschiedener Benutzer-Profil-Felder ermöglicht. Mit dem Entwickeln eines neuen Filter-Handlers und dem Überschreiben des bis dahin verwendeten Filter-Handlers kann «Views» für das Erstellen der Suchlisten verwendet werden. Es ist somit sehr einfach, eine neue Suchliste zu erstellen oder bereits existierende Suchlisten zu bearbeiten. Unterschiedliche, aber inhaltlich übereinstimmende Benutzer-Profil-Felder<sup>6</sup> können ebenfalls gematcht werden. Dies wird durch die Angabe eines Wertes, der entweder ignoriert oder zusätzlich betrachtet werden soll, ermöglicht.

<sup>6</sup>Wenn beispielsweise ein Au-pair als Geschlecht «männlich» angegeben hat, wird bei den Gastfamilien nach «männlich» und «egal» gesucht.

## 3.2 Recommendations

Um in einem Online-Portal Recommendations anzuzeigen, gibt es vier Möglichkeiten: zwei statische und zwei dynamische.

Im Folgenden werden die beiden statischen Möglichkeiten erläutert: Die erste (statische) Variante besteht darin, dass der Administrator das gesamte Geschehen im Portal überwacht und bei jeder Neuansmeldung eines Benutzers dauerhafte Empfehlungen zu diesem selbst definiert. Der Aufwand dieser Methode wäre jedoch so enorm, dass diese Möglichkeit nicht weiter in Betracht gezogen werden kann. In der zweiten statischen Variante würde der Administrator für ihn sehr attraktive Profile als Empfehlung für alle anderen Benutzer auswählen. Durch die vielen Neuansmeldungen und das ständige Ändern der Benutzerprofilstatus (inaktiv, gesperrt, gelöscht) ist auch hier der Aufwand nicht abzuschätzen und somit keine sinnvolle Umsetzung für das Anzeigen von Empfehlungen. Festzuhalten ist, dass beide statischen Möglichkeiten nicht effektiv – weil zu aufwendig – sind. Deshalb sollen nun die dynamischen Varianten betrachtet werden.

Die erste dynamische Lösung wäre das selbstständige Entwickeln eines Recommendation-Systems, welche zwei Vorteile hat:

- eigene Anpassung des Berechnungsansatzes
- Importschnittstelle kann an eigene Bedürfnisse angepasst werden

Neben diesen Vorteilen treten jedoch auch folgende Nachteile bei einer Eigenentwicklung auf:

- enormer Zeitaufwand für die Entwicklung des eigenen Systems
- großes KnowHow nötig, um Berechnungsansatz vorteilhaft zu wählen
- Server zur Berechnung der Recommendations

Der zweite dynamische Ansatz umfasst das Nutzen einer Dienstleistung. Die Vorteile dafür sind:

- Recommendation-System ist bereits entwickelt und sofort einsatzbereit
- um Wartung und Betrieb des Servers muss sich nicht gekümmert werden
- fachkundiges Team kann zur Beratung hinzugezogen werden
- Beziehen von eventuellen Verbesserungen und Erweiterungen möglich
- großes KnowHow über die Berechnungsmöglichkeiten vorhanden

Jedoch gibt es auch hier einige Nachteile:

- Anpassungen an die Berechnung muss mit dem Dienstleister abgesprochen werden
- Importschnittstelle ist vorgegeben
- beschränkte Integrationsmöglichkeiten

Zusammenfassend ist zu sagen, dass bei der Recherche nach Servicepartnern ein Partner gefunden wurde, der für die Integration von Recommendations bereits ein Modul entwickelt hat. Die Importschnittstelle war zwar vorgegeben, da aber der Export der Daten noch nicht entwickelt war, konnte dieser nach den Vorgaben für den Import erstellt werden.

Es wurde sich demzufolge auf die Lösung geeinigt, dass ein Servicepartner die Recommendations bereitstellt.

### 3.2.1 Konzept

Um möglichst schnell Ergebnisse zu erhalten, wurde also auf die Entwicklung eines Recommendation-Systems verzichtet und die bestehende Lösung des Servicepartners

eingesetzt. Der Analytics-As-A-Service (kurz A3S) Recommendation-Service kann historische und interaktive Daten zur Berechnung von Empfehlungen verarbeiten (vgl. Langner, 2010, 2). „Zu den historischen Daten gehören Produktdaten, Transaktionsdaten und Kundendaten.“ (ebd.). Die historischen Daten werden unabhängig von den interaktiven Daten gesammelt. Diese interaktiven Daten werden von einem weiteren A3S-Service, dem A3S-Event-Tracking-Service, erhoben. Er ermittelt beispielsweise Daten über die Verweildauer von Benutzern auf einer Seite, das Anklicken oder Löschen von Empfehlungen durch den Benutzer (vgl. dazu ebd.).

Die Mindestvoraussetzungen für Empfehlungen durch den A3S sind das Bereitstellen der aktuellen Produktdaten durch einen Export sowie die Konfiguration des vom A3S zur Verfügung gestellten Drupal-Moduls. Die exportierten Produktdaten müssen für die Berechnung in den Recommendation-Service importiert werden. Das Importieren kann entweder über einen Pull oder über einen Push realisiert werden. Bei einem Push werden die Daten über eine Schnittstelle direkt an den Server gesendet und importiert. Bei einem Pull legt man die für die Berechnung wichtigen Daten ab, sodass diese vom Service-Anbieter geholt und importiert werden können. Da A3S keine Schnittstelle für das Importieren von Produktdaten anbietet, wurde sich auf das Pull-Prinzip geeinigt.

Das A3S-Team hat ein Drupal-Modul entwickelt, um die Integration der berechneten Recommendations zu erleichtern. Dieses A3S-Modul «A3S Recommendations» stellt nach der notwendigen Konfiguration einen Block bereit, welcher die nach der Berechnung übertragenen Profil-IDs in einem HTML-Template zurückgibt. Dieses Template allerdings bietet – wie in Abbildung 3.7 zu sehen ist – sehr wenig Informationen über die angezeigte Empfehlung, sodass an dieser Stelle Anpassungen notwendig wurden.






- Aigerim J.  (X)
- Wanjiru H.  (X)
- Dinara T.  (X)
- Aisha L.  (X)
- Ann G.  (X)
- marie h.  (X)

Abbildung 3.7: Anzeige von Recommendations mit Standard-HTML-Template

Welche Anpassungen an dem HTML-Template gemacht werden müssen, hängt sehr von der Stelle der Anzeige der Recommendations ab. Eine Möglichkeit wäre es, die Empfehlungen als Suchliste zu verarbeiten. In dem Portal sind bereits Suchlisten vorhanden, sodass sich – um den Benutzer nicht zu verunsichern – an die existierenden Designs dieser Suchlisten gehalten werden muss. Eine weitere Variante, die Empfehlungen in dem Portal einfließen zu lassen, wäre das Anzeigen von «klassischen Recommendations» in der selben Art und Weise, wie es auch Amazon oder ebay machen. Bei den «klassischen Recommendations» beziehen sich die als Empfehlungen angezeigten Produkte auf das Produkt, welches gerade betrachtet wird. Da sich diese Methode in der Praxis bewährt hat, sollen diese Recommendations als «klassisch» bezeichnet werden. Im Vergleich dazu ist es eher unüblich, bei einer Suchliste die empfohlenen Produkte auf den Benutzer, der gerade angemeldet ist, zu beziehen. Dies kann dazu führen, dass es dem Benutzer in diesem Fall vermutlich nicht ersichtlich ist, warum genau diese Profile in der Liste stehen.

Die erste Variante, aus berechneten Empfehlungen eine geeignete Suchliste zu generieren, wurde verworfen, da eine individualisierte Suchliste bereits mit dem Matchmaking realisiert worden ist (vgl. Abschnitt 3.1). Die zweite Variante eignet sich hervorragend, da mit dem Anzeigen von klassischen Empfehlungen, die Usability des Portales weit mehr gesteigert werden kann als mit dem Anzeigen einer durch Empfehlungen generierten Liste. Außerdem spiegeln klassische Empfehlungen die Grundsätze der Usability perfekt wider. Durch die weitverbreitete Anwendung von klassischen Recommendations kann der Benutzer mit diesen umgehen und die Anzeige der Profile



hier viel besser nachvollziehen. Aufgrund dieser Erkenntnisse entschied man sich für die klassischen Recommendations.

### 3.2.2 Implementierung

#### 3.2.2.1 Konfiguration des A3S-Moduls

Die Konfiguration des A3S-Moduls ist simpel. Zunächst gibt man den Inhaltstyp an, auf dem Recommendations eingeblendet werden sollen. Die darauffolgenden drei Einstellungen werden vom A3S-Team vorgegeben. Diese Einstellungen sind der A3S-Host, die Host-API-Version und der Application-Key. Als nächstes muss man die Anzahl der anzuzeigenden Empfehlungen angeben. Um die Einstellungen abzuschließen, wird aus den vorhandenen CCK-Feldern eines ausgewählt, welches als Produkt-ID dient. Nach dieser Konfiguration wird ein Block bereitgestellt, der die Anzeige der Recommendations mit dem Standard-HTML-Template realisiert.

#### 3.2.2.2 HTML-Template-Anpassungen

Dieses Template beschränkt sich bei der Anzeige aber lediglich auf den Titel. Der Titel dient als Link, um zu der entsprechenden Node zu gelangen. Das «X» hinter dem Titel ist Teil des A3S-Event-Tracking-Services (Abbildung 3.7) und dient zum Löschen der angezeigten Recommendation.

Die alleinige Anzeige des Namens eines Au-pair / einer Gastfamilie ist vollkommen unzureichend, um eine passende Auswahl zu treffen. Der Name ist irrelevant, viel hilfreicher wären hier eine Angabe über das Herkunftsland und eventuell den Sprachkenntnissen. Somit wird das Template so angepasst, dass die jeweiligen Empfehlungen aussagekräftiger werden. Templates können bei Drupal in der «template.php» überschrieben werden. Um mehr Informationen zu den angezeigten Profilen zu haben, wurde sich darauf geeinigt, neben dem Namen, das Avatarbild, das Heimatland sowie das Alter der Au-pair in das Layout der Empfehlungen einfließen zu lassen. Bei

der Anzeige der Gastfamilien wird anstatt des Alters die Heimatstadt der Familie gezeigt. Da selbst diese Angaben für eine qualitativ gute Wahl nicht ausreichen, wurde ein CSS-Tooltip verwendet, um bei einem Mouseover mehr Informationen über das Au-pair / die Gastfamilie anzugeben. Ein CSS-Tooltip ist eine clevere Art, dynamischen Inhalt in HTML anzuzeigen, ohne JavaScript zu verwenden (vgl. psacake, 2011, Tooltip, übersetzt von DG).

Der folgende Quelltext-Abschnitt beschreibt, wie man an die darzustellenden Informationen der Benutzer kommt:

Listing 3.10: Benutzer-Felder laden

```
1 $field_home = content_fields('field_home', $node->type);  
2 $field_home['display_settings']['label']['format'] = 'inline';  
3 $field_home_formatted = content_view_field($field_home, $node, FALSE,  
    FALSE);
```

Bei der Umsetzung des neuen Templates mussten zunächst alle anzuzeigenden Felder geladen werden. Die CCK-Methode «content\_fields» gibt eine Liste aller Felder zurück, die ein Inhaltstyp beinhaltet. Wird dieser Funktion – wie in Zeile 1 zu sehen – der Name des Feldes übergeben, so werden nur die Informationen dieses Feldes zurückgegeben. Mit dem zweiten Parameter kann man zwischen den Inhaltstypen differenzieren. Danach können feldspezifische Formatierungen vorgenommen werden. In Zeile 2 wird beispielsweise das Format des Labels – also die Beschriftung – auf «inline» gesetzt. Dies bewirkt, dass das Label vor dem Wert des Feldes auf einer Zeile steht. Das Label steht ohne diese Änderung als Überschrift über dem Feldwert. Um sicherzustellen, dass sich die Recommendations in das vorhandene Layout der Profile einfügen, muss das Feld noch gerendert<sup>7</sup> werden. Die Funktion «content\_view\_field» realisiert diese Anforderung. Auf identische Weise wurden folgende Gastfamilien-Felder geladen, formatiert und gerendert.

---

<sup>7</sup>Bezeichnet die Nachbearbeitung des Feldes. Hier werden Layout, Schriftgröße und Label des Feldes gesetzt. Darüberhinaus wird die Darstellung von multiplen Werten des Feldes festgelegt (vgl. Drupalcontrib, 2011a).

Da sich die Namen der Felder im Quelltext von den Namen der Felder im System unterscheiden, stellt die nachstehende Tabelle die jeweiligen Bezeichnungen gegenüber:

Feldname im System	Feldname im Quelltext
field_avatar	img_formatted
field_gender_search	field_gs_formatted
field_age	field_age_formatted
field_how_much_childs	field_kidsnum_formatted
field_age_childs	field_kids_formatted

Tabelle 3.1: Gegenüberstellung System-Feldname zu Quelltext-Feldname

Eine Gegenüberstellung dieser Gastfamilien-Felder und der Verwendung im System ist im Anhang zu finden (Tabelle A.3). Im Folgenden wird gezeigt, wie diese Felder für den Aufbau des Recommendation-Templates genutzt werden:

Listing 3.11: Benutzer-Felder im Recommendation-Template benutzen

```

1 $output .= l($img_formatted .
2 '<span class="tooltip">
3   <span class="top"></span>
4   <span class="middle">
5     ' . '<b>' . $node->field_firstname[0]['value'] . ' ' . $node->
6       field_lastname_one_char[0]['value'] . ' ' . '</b><br>'
7     . '<hr>'
8     . $field_gs_formatted
9     . $field_age_formatted
10    . $field_kidsnum_formatted
11    . $field_kids_formatted
12    . '
13  </span>
14  <span class="bottom"></span>
15 </span>', 'node/' . $node->nid, array('html' => TRUE, 'query' => array(
16   'rcn' => '1', 'rcnpos' => $pos), 'attributes' => array('class' =>
17   'rcn-tooltip')));
18 $output .= l(t('Family ') . check_plain($node->field_lastname_one_char
19   [0]['value'] . ' ')) .
20 '<span class="tooltip">
21   <span class="top"></span>
22   <span class="middle">

```

```

19      ' . '<b>' . $node->field_firstname[0]['value'] . ' ' . $node->
        field_lastname_one_char[0]['value'] . ' . ' . '</b><br>'
20      . '<hr>'
21      . $field_gs_formatted
22      . $field_age_formatted
23      . $field_kidsnum_formatted
24      . $field_kids_formatted
25      . '
26      </span>
27      <span class="bottom"></span>
28 </span>', 'node/' . $node->nid, array('html' => TRUE, 'query' => array(
        'rcn' => '1', 'rcnpos' => $pos), 'attributes' => array('class' =>
        'rcn-tooltip')));
29 $output .= $field_home_formatted;
30 $output .= $field_city_formatted;
31 }

```

Die Hauptmerkmale, die über eine Familie in einer Empfehlung angezeigt werden, sind das Avatarbild (Zeile 1), das Kürzel «Family» (Zeile 15) und der erste Buchstabe des Nachnamens der Familie – abgeschlossen mit einem Punkt (Zeile 15), dem Heimatland (Zeile 29) und der Heimatstadt (Zeile 30). Die übrigen Informationen werden in dem CSS-Tooltip angezeigt.

Der CSS-Tooltip wird zweimal erstellt. Die Zeilen 2-13 erstellen den CSS-Tooltip, der angezeigt wird, wenn der Benutzer mit der Maus über dem Avatarbild ist. Die Zeilen 16-26 erzeugen den zweiten Tooltip, der erscheint, sobald der Benutzer mit der Maus über dem Kürzel «Family» und dem ersten Buchstaben des Nachnamens ist.

Die Idee eines CSS-Tooltips ist es, in einen vorhandenen Link-Tag «<a>» – der mit der l-Funktion (Zeile 1) generiert wird – zwischen einem öffnenden und schließenden span-Tag «<span> bzw. </span>» zusätzliche Informationen zu setzen. Der CSS-Tooltip besteht aus vier span-Elementen. Das erste Element stellt den kompletten Bereich dar und beinhaltet drei weitere Elemente. Diese werden mit Hilfe der CSS-Klassen «top», «middle» und «bottom» differenziert. Somit ist es später möglich, die einzelnen Teile des CSS-Tooltips mittels CSS unterschiedlich zu formatieren. Das mittlere span-Element «middle» enthält die zusätzlichen Informationen zu dem angezeigten Profil und ist wie folgt aufgebaut:

Eine Überschrift, bestehend aus dem Vornamen, gefolgt von dem ersten Buchstaben des Nachnamens, leitet die Zusatz-Informationen ein. Danach wird mit dem HTML-Tag «<hr>» eine horizontale Linie erstellt (Zeile 6). Nun werden die Gastfamilien-Felder in das HTML-Template eingefügt, auf die sich im Vorfeld geeinigt wurde (Zeile 7, 8, 9 und 10).

Diese span-Tags werden anschließend mittels CSS versteckt:

Listing 3.12: CSS: span-Tag verstecken

```
1 a.rcn-tooltip span {  
2     display: none;  
3 }
```

Diese CSS-Anweisung setzt die Eigenschaft «display» auf den Wert «none». Dadurch werden alle span-Elemente in einem Link <a>, der die CSS-Klasse «rcn-tooltip» besitzt, versteckt. Das heißt der span-Tag und die darin enthaltenen Informationen werden nicht angezeigt. Erst bei dem Ereignis «hover» (Zeile 1 des Listings 3.13), welches angibt, dass der Mauszeiger auf dem Link ist, werden die span-Tags und somit die Informationen wieder sichtbar. Der folgende Quelltext veranschaulicht dies:

Listing 3.13: CSS: Recommendation-Anzeige formatieren

```
1 a.rcn-tooltip:hover span.middle {  
2     background: none repeat scroll 0 0 white;  
3     border-left: 1px solid #D9EAF5;  
4     border-right: 1px solid #D9EAF5;  
5     display: block;  
6     padding: 0 8px;  
7 }
```

Der span-Tag mit der CSS-Klasse «middle» (Zeile 1) bekommt einen weißen Hintergrund (Zeile 2), rechts und links einen Rahmen, der einen Pixel stark ist und eine durchgezogene Linie sowie die Farbe «#D9EAF5» (Hellblau) hat (Zeile 3, 4). Danach wird die zum Anzeigen wichtige Eigenschaft «display» auf den Wert «block» gesetzt (Zeile 5).

Die Implementierung und Formatierung des CSS-Tooltips ist somit abgeschlossen.

In der Abbildung 3.8 werden Recommendations und ein dazugehöriger CSS-Tooltip gezeigt:

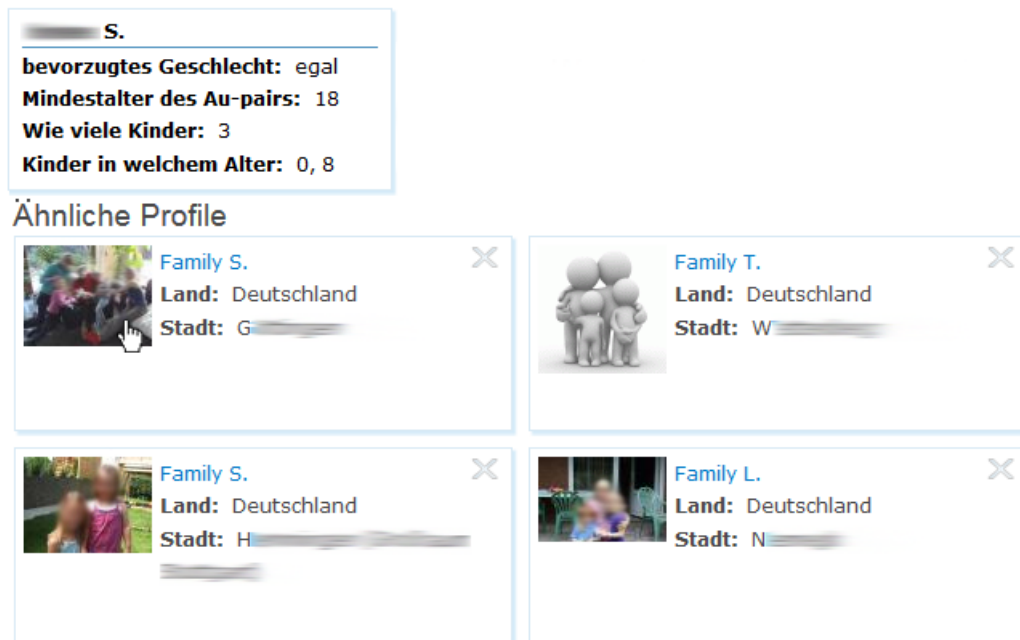


Abbildung 3.8: Anzeige von Recommendations und Tooltip mit angepasstem HTML-Template

In den folgenden Abschnitten wird der Ablauf erläutert, wie die Benutzerdaten zum A3S gelangen und welche Filterung für qualitativ hochwertige Empfehlungen notwendig sind. Die Abbildung 3.9 beschreibt den Prozess der Empfehlungsanzeige:

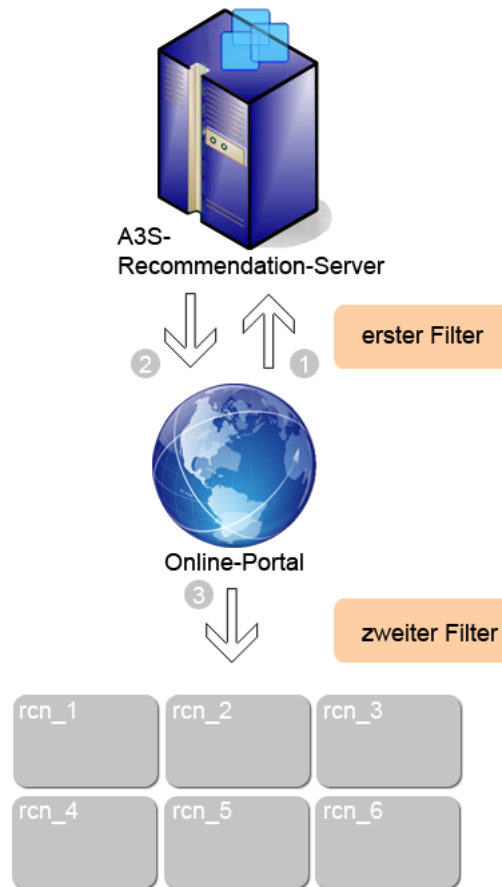


Abbildung 3.9: Ablauf der Filterung vor der Anzeige der Empfehlungen

Die Pfeile sind der Reihe nach nummeriert und geben somit eine Reihenfolge an. Nummer 1 beschreibt den Export der Benutzerdaten. An dieser Stelle wird bereits ein erster Filter eingesetzt, um keine unnötigen Daten zu berechnen. Der Pfeil mit der Nummer 2 symbolisiert, wie die berechneten Daten dem Portal wieder zurückgeliefert und bearbeitet werden. Diese Funktionalität stellt das zur Verfügung gestellte A3S-Modul bereit, daher wird in dieser Arbeit nicht näher darauf eingegangen. Pfeil Nummer 3 stellt das Ausliefern der berechneten Empfehlungen auf das HTML-Template dar. An dieser Stelle wird ein zweiter Filter eingesetzt, um die Qualität der Empfehlungen zu verbessern. Im nächsten Abschnitt wird der Datenexport genauer betrachtet:

### 3.2.2.3 Datenexport

Beim Liefern der Benutzerdaten muss sich an die vom A3S-Team gestellten Vorgaben gehalten werden. Dabei handelt es sich um eine CSV-Datei. Auch das Format dieser Datei wird von A3S vorgegeben. In der Kopfzeile der Datei sollen die Feldnamen stehen. Die Feldnamen und die Werte der einzelnen Felder werden zwischen Hochkommas gesetzt. Hat ein Feld mehrere Werte, so werden diese durch ein Komma getrennt. Der Trenner der einzelnen Felder ist ein Semikolon. Als Beispiel sieht man in Abbildung 3.10 die ersten sechs Zeilen der Export-CSV-Datei, wie sie im Portal exportiert wird.

```
"nid";"Typ";"gender";"home_aupair";"home_family";"good_lang"
"224";"Au-pair";"1";"45";"41,14,47";"Deutsch,Russisch,Andere Sprache"
"256";"Au-pair";"1";"45";"13,40,16,46";"Deutsch,Russisch"
"298";"Au-pair";"1";"60";"41,14";"Russisch"
"326";"Au-pair";"1";"60";"14";"Russisch"
"330";"Au-pair";"1";"18";"41,17,46";"Englisch,Deutsch,Russisch"
"332";"Gastfamilie";"1";"4,6,8,10,13,20,28,31,38,42,43,44,45,49,50,56,60,61,62,66";"14";"Deutsch"
```

Abbildung 3.10: Struktur der CSV-Datei

In der ersten Zeile stehen – wie es von A3S vorgegeben war – die Feldnamen. «nid» ist dabei die Node-ID, welche Drupal beim Erstellen der Benutzerprofile fortlaufend vergibt. Der Recommendation-Service nutzt diese Node-ID als Produkt-ID. Das Feld «Typ» beinhaltet den Benutzerprofil-Typ. Das Feld «gender» enthält die Werte des Geschlechtes. Bei Au-pair ist dies die Angabe zu ihrem Geschlecht und bei Gastfamilien ist es die Angabe, welches Geschlecht das Au-pair haben soll. Wie in der Abbildung 3.10 zu sehen ist, hat jede Zeile (bis auf die Kopfzeile) an der dritten Stelle den Wert «1» und nicht etwa Angaben wie «männlich», «weiblich» oder «egal». Diese Zahl ist lediglich ein Key zu einem Wert. In diesem Feld ist mit dem Key «1» der Wert «weiblich» verknüpft. Andere mögliche Keys an dieser Stelle sind in der Tabelle A.1 im Anhang zu finden.

Die Werte in dem folgenden Feld sind ebenfalls vom Benutzerprofil-Typ abhängig: «home\_aupair» beinhaltet zum einen den Wert des Heimatlandes des Au-pair und zum anderen die Angaben, die eine Gastfamilie über das Land, aus dem das Au-pair kommen soll, gemacht hat. In dem Feld «home\_family» verhält es sich genau anders



herum. Hier sind es die Angaben des Au-pair, aus welchem Land die Gastfamilie sein soll und die Angabe des Heimatlandes der Gastfamilie. Beide Felder enthalten wieder nur die Keys der Werte. Eine vollständige Liste der Länder und der dazugehörigen Keys ist ebenfalls im Anhang zu finden (Tabelle A.2). Das letzte Feld «good\_lang» gibt die Werte an, die die Benutzer bei der Frage nach «gute[n] Sprachkenntnisse[n]» angegeben haben.

Die beschriebene Export-CSV-Datei wird aus einer Drupal-View generiert. Um große Datenmengen aus einer Drupal-View zu exportieren, gibt es das Drupal-Modul «Views data export» (Views Data Export, 2011). Damit ist es möglich, die Daten einer erstellten View in eine CSV-, Microsoft XLS-, Microsoft DOC-, Basic TXT- oder XML-Datei zu exportieren. Welche Daten in dieser bereitgestellt werden, hängt von den in der View angegebenen Feldern ab. In Abbildung 3.11 sind die zu exportierenden Felder für das Portal zu sehen:

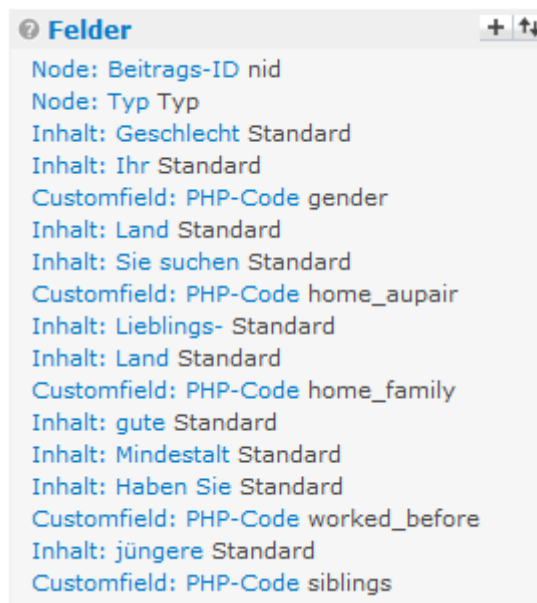


Abbildung 3.11: Felder der Data-Export-View

Die Felder, die Drupal zur Verfügung stellt, sind «Beitrags-ID» und «Typ». Beide sind mit «Node:» gekennzeichnet, da es nodespezifische Angaben sind. Die Felder aus den Benutzerprofilen sind in der Abbildung mit «Inhalt:» gekennzeichnet. Die mit

«Customfield:» ausgezeichneten Felder sind PHP-Code-Felder. Diese Felder dienen beispielsweise dazu, die Angabe der Au-pair über ihr Geschlecht und die Angaben der Gastfamilien über das gesuchte Geschlecht des Au-pair in einem Feld zu vereinen.

Die durch das Modul erzeugte Datei wird allerdings sofort zum Download angeboten. Da sich aber auf das Pull-Prinzip geeinigt wurde, musste die erzeugte Datei an einem bestimmten Ort auf dem Server hinterlassen werden (vgl. Abschnitt 3.2.1). Um diese Funktionalität zu schaffen, arbeitet das Modul mit «Drush»<sup>8</sup> (Drush, 2011) zusammen. Drush ist eine Kommandozeilen-Interface, um Shell- oder Script-Anweisungen schreiben zu können. Drush wird nicht wie andere Drupal-Module in das Drupal-System integriert, sondern auf dem Server installiert. Es muss somit eine Anweisung geschrieben werden, welche die erstellte Data-Export-View ausführt und als Datei auf den Server schreibt. Darüber hinaus muss der Prozess des Datenexports automatisiert ablaufen, da sich die Benutzerdaten und somit die Produktdaten ständig durch Neuanmeldungen, Inaktivsetzen oder Sperren der Profile ändern. Der Export der Daten muss demnach mindestens jeden Tag erfolgen, um aktuelle Empfehlungen zu berechnen. Im Folgenden wird darauf eingegangen, wie dies umgesetzt wurde.

Um immer wiederkehrende Aktionen auf einem Unix-System zu automatisieren, kann man einen «Cronjob»<sup>9</sup> verwenden.

Im nachstehenden Quelltext wird der Cronjob, der diese Aufgabe realisiert, näher betrachtet (Der Cronjob wurde zur Verdeutlichung auf mehrere Zeilen verteilt.):

Listing 3.14: Cronjob

```
1 40 23 * * *
2 cd htdocs/www && mkdir -p rcn_export && cd rcn_export && cp ~/
  rcn_export/.htaccess . &&
3 ~/drush/drush views-data-export admin_lists views_data_export_1
4 export_wap_cb.csv
```

Zeile 1 beinhaltet die Anweisungen für das zeitgesteuerte Ablaufen des Jobs. Dafür stehen fünf Felder zur Verfügung: Minute, Stunde, Tag des Monats, Monat und der Wochentag. Alle Felder können numerische Werte annehmen und der Stern (\*)

---

<sup>8</sup>Der Name «Drush» setzt sich aus Drupal und Shell zusammen.

<sup>9</sup>Cronjob wird verwendet, um wiederkehrende Abarbeitung einer Aufgabentabelle (Crontab) im Betriebssystem des jeweiligen Servers durchzuführen (vgl. Business-Webhosting.de, 2011).

gilt als Platzhalter. Dies bedeutet, dass an dieser Stelle jeder gültige Wert benutzt werden soll. Zeile 1 gibt also an, dass der Job in der 40sten Minute der 23sten Stunde an jedem Tag im Monat, jeden Monat und an jedem Wochentag ausgeführt werden soll – ergo alle 24 Stunden um 23:40 Uhr.

In Zeile 2 wird der Ordner «rcn\_export» erstellt, in dem die Export-Datei gespeichert werden soll. Außerdem wird eine «.htaccess» in diesen Ordner kopiert, um diesen vor unberechtigten Zugriffen zu schützen. Dies wird in Abschnitt 3.2.2.5 genauer erläutert. Zeile 3 zeigt, wie mittels Drush das Kommando «views-data-export» aufgerufen wird. Dem Kommando werden die folgenden Parameter übergeben: Name der View, Display-ID und Name der Output-File. In der View «admin\_lists» wird also das Display «views\_data\_export\_1» aufgerufen und das Resultat in der «export\_wap\_cb.csv» (Zeile 4) auf dem Server gespeichert.

Bevor jedoch die Export-Datei erstellt wird, muss eine Filterung der Daten vorgenommen werden. Wie in Abbildung 3.9 zu sehen ist, wird an dieser Stelle der «erste Filter» eingesetzt. Wie dieser Filter die zu exportierenden Daten bearbeitet, thematisiert der nächste Abschnitt.

### 3.2.2.4 Erster Filter

Filter können bei dem Erstellen der Data-Export-View angegeben werden. Beispielsweise sollten nur Benutzer exportiert werden, welche die Rollen «aktiver nutzer»<sup>10</sup> und «aktivierter Benutzer» haben. Somit wäre dann sichergestellt, dass der Benutzer sich nach der Registrierung mindestens ein Mal im Portal angemeldet und sein Profil nicht inaktiv gesetzt hat. Weiterhin wurden durch die Rollen die Testprofile nicht mitexportiert, indem man nach der Rolle «Testprofile» filtert. Eine wichtige Eingrenzung beim Erstellen der Export-Datei ist folgende: Alle Benutzer, deren letzter Zugriff länger als sechs Wochen her ist, fließen *nicht* mit in die Berechnung ein. Da diese Benutzer womöglich das Interesse an dem Portal verloren haben und es eher unwahrscheinlich ist, sie mit dem Senden einer Kontaktanfrage zu reaktivieren, werden sie nicht berücksichtigt. Darüber hinaus ist es für aktive Benutzer

---

<sup>10</sup>Kleinschreibung, da so im System.

sehr frustrierend, wenn sie von einem solchen Benutzer keine Rückmeldung auf ihre Kontaktanfrage erhalten.

Die Abbildung 3.12 zeigt die in der Data-Export-View angegebenen Filter:



Abbildung 3.12: Filter der Data-Export-View

Die nötige Vorfilterung der Benutzerdaten ist somit abgeschlossen. Nun kann die Export-CSV-Datei mit dem bereits erklärten Cronjob (vgl. Abschnitt 3.2.2.3) erstellt werden. Da diese Datei in einem Ordner auf dem Server gespeichert wird, auf den jeder Zugriff hat, muss eine Sicherung dieses Ordners erfolgen. Das wurde mit einer «.htaccess»-Datei realisiert. Im nächsten Abschnitt wird genauer auf die dafür notwendigen Schritte eingegangen.

### 3.2.2.5 Ordnersicherung mittels .htaccess

Um einen Ordner auf einem Server vor unberechtigten Zugriffen zu schützen, kann der Ordner mit einer «.htaccess»- und der dazugehörigen «.passwd»-Datei gesichert werden. Der Ordner fragt daraufhin bei einem Zugriff nach Benutzernamen und Passwort. Dem A3S-Team wurden Benutzername und Passwort mitgeteilt, sodass sie auf diesen Ordner zugreifen, die aktuelle Datei in ihr System importieren und neue berechnete Ergebnisse ausliefern können. Im Folgenden wird auf die Struktur der «.htaccess» genauer eingegangen:

Listing 3.15: .htaccess-Datei

```
1  # .htaccess-Datei fuer DataExport
2  AuthType Basic
3  AuthName "Data Export"
4  AuthUserFile /server/home/rcn_export/.htpasswd
5  <Files *.csv>
6  Require user a3s
7  </Files>
```

Zeile 1 beinhaltet lediglich einen Kommentar und dient hier als Überschrift in der Datei. Der «AuthType» in Zeile 2 gibt an, um welche Art es sich bei der Authentifizierung handelt. «Basic» steht hier für die oft gebrauchte, sogenannte HTTP Basic Authentication. Diese Methode verlangt zwar, dass die Passwörter und Benutzernamen unverschlüsselt an den Server übertragen werden, eine verschlüsselte Übertragung wird jedoch nicht von allen Browsern unterstützt (vgl. SELFHTML, 2007).

In Zeile 3 wird ein «AuthName» angegeben. Dies ist eine kurze Beschreibung des geschützten Web-Bereichs und wird dem Benutzer bei der Passwortabfrage angezeigt (vgl. Brinkmann, 1998). Mit «AuthUserFile» (Zeile 4) wird der Pfad zu der Datei angegeben, in der die Benutzernamen und deren Passwörter gespeichert sind. Wie eine solche Datei aussehen kann, wird im Anschluss an die «.htaccess» beschrieben. Mittels des «<Files>»-Bereiches, der in Zeile 5 beginnt und in Zeile 7 endet, kann man zwischen den in dem Ordner befindlichen Dateien differenzieren. Man muss also nicht zwingend den gesamten Ordner sichern, sondern kann auch einzelne Dateien oder Gruppen von Dateien sichern. In dem Beispiel hier (Zeile 5) werden alle Dateien mit der Endung «.csv» gesichert. Zeile 6 gibt an, welcher Benutzer oder welche Benutzergruppe die gesicherten Dateien anfordern darf. Im obenstehenden Beispiel ist ersichtlich, dass nur *ein* Benutzer «a3s» mit dem dazugehörigen Passwort auf alle Dateien mit der Endung «.csv» zugreifen darf.

Um die Passwörter der Benutzer zu speichern, ist es nötig, eine .htpasswd-Datei anzulegen und ebenfalls auf dem Server zu speichern. Der folgende Quelltext zeigt, wie eine solche Datei aussehen könnte<sup>11</sup>:

---

<sup>11</sup>Diese Datei ist vollkommen fiktiv und dient lediglich als Verdeutlichung

Listing 3.16: .htpasswd-Datei

```

1 # BenutzerDatei fuer Web-Projekt
2 a3s:fsdafsaghzssa.
3 backup:sadniuggalla.
4 Bernd:Poseidon
5 Karin:Hera

```

Zeile 1 ist – wie schon in der «.htaccess» – ein Kommentar und dient auch hier als Überschrift. Die Zeilen 2, 3, 4 und 5 enthalten die Benutzer und deren Passwörter, getrennt durch einen «:». Es ist möglich, die Passwörter wie in Zeile 2 und 3 verschlüsselt<sup>12</sup> anzuzeigen. Man kann die Passwörter aber auch unverschlüsselt angeben (Zeile 4 und 5), was jedoch ein erhebliches Sicherheitsrisiko darstellt.

Festzuhalten ist, dass der Datenexport damit abgeschlossen ist und die Export-CSV-Datei auf dem Server vor unberechtigten Zugriffen mit Hilfe der «.htaccess»- und der «.passwd»-Datei geschützt wird.

Der A3S-Recommendation-Server kann Empfehlungen berechnen und dem Online-Portal zur Verfügung stellen. Es wäre an dieser Stelle bereits möglich, die Empfehlungen in dem HTML-Template anzuzeigen. Da sich aber – wie bereits beschrieben – die Benutzerdaten ständig ändern und der Export der Daten nur einmal am Tag stattfindet, ist es so, dass bis zum nächsten Export wichtige Profiländerungen – wie zum Beispiel das Inaktivsetzen des Profils oder das Sperren eines Benutzers durch den Administrator – beim Anzeigen der Empfehlungen vollkommen ignoriert werden. Um dieses Problem zu beheben, werden die gelieferten Empfehlungen gefiltert. Abbildung 3.9 bezeichnet diesen Schritt als «zweiter Filter». Auf diesen wird nun detaillierter eingegangen.

### 3.2.2.6 Zweiter Filter

Bei der Analyse der gelieferten Empfehlungen ergaben sich – zusätzlich zu dem bereits genannten Problem des zeitlichen Ablaufs der Berechnung der Recommendations – weitere Probleme, die den Einsatz eines zweiten Filters nötig machten.

<sup>12</sup>Als Verschlüsselungsmethoden werden Crypt und MD5 akzeptiert.

Es wurde festgestellt, dass von dem Recommendation-System zwar zueinander passende Profile errechnet werden, der Typ der Profile allerdings vollkommen außer Acht gelassen wird. Es werden also z. B. zu einem Au-pair auch Gastfamilien empfohlen. Durch die strikte Trennung der Profile im Portal muss hier eine Filterung erfolgen.

Ein portalspezifisches Problem ergab sich aus den bisher gesammelten Erfahrungen mit den im Portal angemeldeten männlichen Au-pair. Bei ihnen ist aufgefallen, dass sie auch Familien kontaktieren, die nur weibliche Au-pair suchen. Da zu beobachten war, dass auch eine abgelehnte Kontaktanfrage die männlichen Au-pair nicht davon abhalten konnte, immer wieder an dieselbe Familie eine Anfrage zu schicken, mussten die Recommendations für diese Benutzergruppe angepasst werden. Die Anforderung, die sich daraus ergab, bestand darin, den männlichen Au-pair nur Familien anzuzeigen, die auf ihrem Profil folgende Angaben gemacht haben: die Gastfamilie sucht nur «männlich» bzw. «egal».

Um diese Probleme zu beheben, wurde das Modul «rcn\_filter» entworfen. Mit Hilfe dieses Moduls wird durch die Implementierung eines Hooks noch vor der Anzeige der berechneten Empfehlungen nach den bereits erwähnten Kriterien gefiltert. Erst dann sind die Empfehlungen soweit aufbereitet, dass sie dem Benutzer angezeigt werden können.

Mit dem Drupal-Hook «hook\_a3s\_rcn\_items», den das A3S-Modul bereitstellt, kann ein Filtern von Empfehlungen umgesetzt werden. Die folgende Quelltextzeile zeigt diesen Hook: <sup>13</sup>

Listing 3.17: Von A3S bereitgestellter Hook zum Filtern der Empfehlungen

```
1 function hook_a3s_rcn_items($items, $config_id, $item_id)
```

Der Hook erwartet drei Parameter. Der erste Parameter «\$items» beinhaltet die berechneten Empfehlungen. Der zweite Parameter «\$config\_id» ist die ID des Blockes, in dem die Recommendations angezeigt werden sollen. Im letzten Parameter «\$item\_id» steht die ID des Benutzers, zu dem Empfehlungen ausgeliefert werden können.

---

<sup>13</sup>Dieser Quelltext dient der Verdeutlichung. Der tatsächliche Quelltext ist auf der CD im Anhang zu finden.

Zunächst muss die strikte Trennung der Profile wieder gegeben sein. Der folgende Quelltext-Abschnitt realisiert diese Anforderung:

Listing 3.18: Unterscheidung der Benutzertypen

```

1      if ($profile != NULL && $profile->type == 'au_pair'
2          && $item['data']->type == 'au_pair'){
3          $not_allowed = TRUE;
4      }
5      if ($profile != NULL && $profile->type == 'family'
6          && $item['data']->type == 'family'){
7          $not_allowed = TRUE;
8      }

```

Am Anfang muss das Profil des angemeldeten Benutzers geladen werden. Zeile 1 zeigt den Abgleich des Profiltyps des angemeldeten Benutzers. Ist dieser Typ gleich mit «au\_pair», wird in Zeile 2 mit dem Typ des in der Empfehlung angezeigten Benutzers verglichen. Sollte auch dieser Typ den Wert «au\_pair» besitzen, so wird in Zeile 3 die Variable «not\_allowed» auf TRUE gesetzt. Äquivalent ist das Vorgehen für den Benutzertyp «family» - wie in Zeile 5, 6 und 7 zu sehen ist.

Darauffolgend wird von den angezeigten Profilen in den Empfehlungen eine Profileinstellung überprüft:

Listing 3.19: Benutzer auf inaktiv überprüfen

```

1      if ($item['data']->field_profile_settings[0]['value'] == 1) {
2          $set_inactive = TRUE;
3      }

```

Der in dem Feld «field\_profile\_settings» gesetzte Wert gibt an, ob ein Benutzer sein Profil inaktiv gesetzt hat oder nicht. Diese Veränderung eines Profiles innerhalb eines Tages, welche – durch die Tagesaktualität – noch nicht vom Recommendation-System erfasst wurde, kann somit ausgewertet werden. Die Variable «set\_inactive» (Zeile 2) wird gesetzt und kann später zum Filtern der Empfehlungen benutzt werden.

Das spezifische Problem mit den männlichen Au-pair wird mit dem folgenden Quelltext gelöst:



Listing 3.20: Benutzer auf männlich überprüfen

```

1      if ($profile != NULL &&
2          $profile->type == 'au_pair' &&
3          $profile->field_gender[0]['value'] == 2 &&
4          $item['data']->type == 'family' &&
5          $item['data']->field_gender_search[0]['value'] == 1) {
6          $not_allowed = TRUE;
7      }

```

Um Fehler zu vermeiden, wird in Zeile 1 zuerst überprüft, ob die Variable «\$profile» nicht leer ist. In Zeile 2 wird der Profiltyp auf «au\_pair» überprüft. Zeile 3 liest den in dem Feld «field\_gender» gesetzten Wert aus und überprüft diesen auf 2 – also männlich. Nach diesen beiden Abfragen ist klar, dass es sich bei dem angemeldeten Benutzer um ein männliches Au-pair handelt. Daraufgehend wird in Zeile 4 der Typ des in der Empfehlung angezeigten Profiles auf «family» überprüft. Ist diese Aussage wahr, so wird in Zeile 5 nach dem in dem Feld «field\_gender\_search» gesetzten Wert gesucht. Ist dieser Wert 1, dann sucht die Familie explizit nach weiblichen Au-pair. Die Variable «\$not\_allowed» wird auf TRUE gesetzt und kann später ausgewertet werden (Zeile 6).

Zuletzt müssen auch gesperrte Benutzer aus den Empfehlungen entfernt werden. Benutzer, die gesperrt wurden, bekommen bei Drupal den Status 0. Es musste somit nur auf diesen Status überprüft werden, um herauszubekommen, ob ein Benutzer gesperrt ist oder nicht<sup>14</sup>:

Listing 3.21: Löschen nicht erlaubter Recommendations

```

1      if ($rcn_user->status != 1 || $set_inactive || $not_allowed) {
2          unset($items[$index]);
3      }

```

Die Empfehlungen werden daher mit «unset» (Zeile 2) gelöscht, wenn die Überprüfung auf gesperrt, «set\_inactive» oder «not\_allowed» TRUE zurückliefert. Die Empfehlungen sind nun soweit aufgearbeitet, dass sie durch das HTML-Template auf den Profilen angezeigt werden können.

<sup>14</sup>Im Quelltext wird auf den Umkehrschluss – Benutzer hat *nicht* Status 1 – überprüft.

### 3.2.3 Fazit

Abschließend ist zu sagen, dass ein automatisierter Prozess für die Anzeige von berechneten Empfehlungen entwickelt wurde. Zwei Filter verarbeiten hierbei die Daten so, dass keine fehlerhaften Profile in den Empfehlungen zu sehen sind. Ein Filter verarbeitet die Daten noch bevor diese als Export-Datei für den A3S-Service-Partner zur Verfügung stehen. Ein anderer Filter verarbeitet die berechneten Daten des A3S-Service-Partners, wodurch es möglich ist, auf zwischenzeitliche Profiländerungen zu reagieren. Erst nachdem diese beiden Filter durchlaufen worden sind, werden die Empfehlungen mit dem dafür überschriebenen HTML-Template auf den Benutzerprofilen angezeigt. Der Betreiber der Website hat dem Benutzer eine Möglichkeit geschaffen, wie er auf zusätzliche Profile aufmerksam gemacht werden kann. Desweiteren hat der Betreiber keinen Wartungsaufwand beim Ablauf dieses Schemas.

## 3.3 A/B-Test

Eine Anforderung an das Portal war es, eine Umgebung zu schaffen, mit der es möglich ist, A/B-Tests durchzuführen. Das A3S-Team ist aufgrund unterschiedlicher Berechnungsansätze in der Lage, verschiedene Empfehlungen zu generieren. Diese sollten im Portal gegeneinander getestet werden, um die Empfehlungen zu verbessern. Verbesserte Empfehlungen sprechen wiederum für eine Erhöhung der Usability.

### 3.3.1 Konzept

Die Empfehlungen werden auf den Profil-Seiten der Au-pair / Gastfamilien dargestellt. Um einen A/B-Test dieser Empfehlungen durchführen zu können, muss man in der Lage sein, Benutzer in Gruppen einzuteilen und dann je nach Gruppe unterschiedliche Empfehlungen auf den Profil-Seiten einzublenden. Bei einem A/B-Test ist es nicht vorgegeben, dass man die Gruppen immer in gleich große Teile zerlegt. Ergo sollte man hier die Option haben, einen prozentualen Wert für die Teilung angeben zu

können.

Aufgrund des ungleichmäßigen Verhältnisses von Au-pair zu Gastfamilien im Portal wurde die Anforderung an die A/B-Test-Umgebung gestellt, dass die Benutzer anhand ihrer Rolle geteilt und erst dann in die Gruppen eingegliedert werden. Die Menge aller Au-pair sowie Gastfamilien soll also getrennt voneinander unterteilt werden.

Es ist davon auszugehen, dass mit *einem* A/B-Test nicht alle Varianten der Berechnungsansätze getestet werden können. Somit sollte das Unterscheiden verschiedener Tests mittels ID gegeben sein, um weitere Tests zu implementieren.

### 3.3.2 Implementierung

Um einen A/B-Test durchzuführen, ist es notwendig, die Benutzer in verschiedene Gruppen zu teilen. Das Modul «user\_splitting» realisiert diese Aufgabe. Desweiteren stellt das Modul Abfragefunktionen bereit, um die Zugehörigkeit eines Benutzers zu einer Gruppe zu überprüfen. Diese Funktionen können später an geeigneter Stelle verwendet werden, um die Benutzer für den A/B-Test zu unterscheiden. Im Abschnitt 3.3.2.3 wird darauf eingegangen, wie diese Funktionen in einem Panel benutzt werden. Für das Modul «user\_splitting» wurde eine Einstellungsmaske programmiert, um die Grundeinstellungen für einen A/B-Test vorzunehmen. In den folgenden Abschnitten wird genauer auf das Modul eingegangen, indem die Einstellungsmaske mit Hilfe eines Bildes erklärt wird. Die Funktionalität hingegen wird anhand des Quelltextes beschrieben.

#### 3.3.2.1 Konfiguration des user\_splitting-Moduls

Wie in Abbildung 3.13 zu sehen ist, können in der Einstellungsmaske des Moduls user\_splitting die Grundeinstellungen vorgenommen werden:

**User Splitting**

**Welche Benutzergruppen sollen am A / B -Test teilnehmen?:**

- ☐ Authentifizierter Benutzer
- ☐ agentur
- ☐ aktiver nutzer
- ☐ aktivierter Benutzer
- ☒ Au-Pair
- ☒ Gastfamilie
- ☐ old\_user
- ☐ premium
- ☐ testprofile
- ☐ web-aupair-team

**Test ID:**

1

**Wieviel Prozent der Benutzer sollen in Gruppe B:**

50

Gruppe A ist die Benutzergruppe ohne Änderungen, die Gruppe B hat die neue Variante

Abbildung 3.13: user\_splitting-Einstellungsmaske

Zunächst wählt man die zu dem A/B-Test hinzuzufügenden Benutzergruppen. Diese Liste stellt alle im Drupal-System bekannten Rollen dar. Werden Änderungen an den Rollen vorgenommen oder werden neue hinzugefügt, so aktualisiert sich diese Liste automatisch. Die nächste Einstellung realisiert das Nummerieren der Tests, um immer wieder neue Tests definieren zu können. In der dritten Textbox kann man einen Wert angeben, der angibt, wieviel Prozent der Benutzer in Gruppe B eingeteilt werden sollen. Die Gruppe B ist die Gruppe, mit der man die Neuerungen im Portal testet. Gruppe A dient demnach als Referenzgruppe. In der Abbildung 3.13 sind dies 50, das heißt 50% der Benutzer werden in die Gruppe A und 50% in die Gruppe B eingeteilt.

### 3.3.2.2 Implementierung der Funktionalität

Die Benutzer werden, sofern sie noch keiner Gruppe angehören, bei dem Login in das Portal in eine der beiden Gruppen eingeteilt. Außerdem wird durch eine if-Verzweigung verhindert, dass ein Benutzer mehrmals eingeteilt wird. Durch die Teilung bei dem Login entstehen zwei Vorteile: Man hat erstens keine Benutzer in einer Gruppe, die sich zwar registrieren, aber sich danach nie in das Portal einloggen, und man kann zweitens auch Benutzer den Gruppen zuteilen, die schon längere Zeit im Portal registriert sind.

Um eigene Funktionalität an den Login-Prozess von Drupal anzuhängen, wird der von Drupal bereitgestellte Hook «user» verwendet. Der nachfolgende Quelltext zeigt, wie sich an den Login-Prozess gegangen wird:

Listing 3.22: Benutzer bei Login überprüfen

```

1 function user_splitting_user($op, &$edit, &$account, $category = Null)
2   {
3     if ($op == "login") {
4       $uid = $account->uid;
5       $default_values = user_splitting_settings_get_default_values();
6       $test_id = variable_get('user_splitting_test_id',
7                             $default_values['user_splitting_test_id']);
7       $is_assigned = db_fetch_object(db_query("SELECT * FROM {
8         user_splitting} WHERE uid = '%d' AND test_id = '%d'", $uid,
9         $test_id));
10      if ($is_assigned == FALSE){
11        user_splitting_user_assign($account, $test_id);
12      }
13    }
14  }

```

Nachdem der Funktion als Operator der Wert «login» übergeben wurde (Zeile 3), wird von dem Benutzer, der sich gerade angemeldet hat, die UID geladen (Zeile 4). Sobald aus den Konfigurationseinstellungen des Moduls die «test\_id» geladen wurde (Zeile 6), wird ein SQL-Statement ausgeführt (Zeile 7). Dieses SQL-Statement überprüft, ob der Benutzer mit jener UID und der angegebenen Test-ID in der Datenbanktabelle «user\_splitting» existiert oder nicht. Sollte der Benutzer nicht

in der Tabelle sein, so heißt dies, dass er dem Test noch nicht zugeordnet wurde. Die Variable «\$is\_assigned» gibt dann FALSE zurück (Zeile 8). Es wird der Wert dieser Variable überprüft und bei FALSE die Methode «user\_splitting\_user\_assign» aufgerufen (Zeile 9), welche für das Einteilen der Benutzer in die Gruppen zuständig ist.

Im folgenden Quelltext wird die Methode für das Einteilen der Benutzer in die Gruppen genauer betrachtet:

Listing 3.23: Benutzer in Testgruppen einteilen

```

1 function user_splitting_user_assign($account, $test_id) {
2   $user_has_selected_role = FALSE;
3   $default_values = user_splitting_settings_get_default_values();
4   $selected_roles = variable_get('user_splitting_user_types',
      $default_values['user_splitting_user_types']);
5   foreach ($selected_roles as $role_id) {
6     if (array_key_exists($role_id, $account->roles)){
7       $user_has_selected_role = $role_id;
8     }
9   }
10
11  $var_zero_count = db_result(db_query("SELECT COUNT(*) FROM {
      user_splitting} JOIN {users_roles} ON user_splitting.uid =
      users_roles.uid WHERE users_roles.rid = '%d' AND user_splitting.
      var = 0", $user_has_selected_role));
12  $var_one_count = db_result(db_query("SELECT COUNT(*) FROM {
      user_splitting} JOIN {users_roles} ON user_splitting.uid =
      users_roles.uid WHERE users_roles.rid = '%d' AND user_splitting.
      var = 1", $user_has_selected_role));
13
14  $percent_value = variable_get('user_splitting_percent_value',
15    $default_values['user_splitting_percent_value']);
16
17  $sum = $var_zero_count + $var_one_count;
18  $calc_threshold = (($sum * $percent_value) / 100);
19    if ($user_has_selected_role !== FALSE) {
20    if ($var_one_count < $calc_threshold) {
21      $var = 1;

```

```

22     db_query("INSERT INTO {user_splitting} (uid, test_id, var) VALUES
        (%d, %d, %d)", $account->uid, $test_id, $var);
23 } else {
24     $var = 0;
25     db_query("INSERT INTO {user_splitting} (uid, test_id, var) VALUES
        (%d, %d, %d)", $account->uid, $test_id, $var);
26 }
27 }
28 }

```

Da es – wie bereits im Abschnitt 3.3.2.1 erwähnt – möglich ist, mehrere Rollen am A/B-Test teilhaben zu lassen, wird zunächst in Zeile 6 überprüft ob der Benutzer die Rolle besitzt, welche am A/B-Test teilnehmen soll. Das SQL-Statement in Zeile 11 zählt dann die Benutzer, die in der Datenbanktabelle `user_splitting` stehen und der Variante 0 (`var = 0`) angehören. Dies wird in der Variable «`$var_zero_count`» gespeichert. Mittels JOIN wird vorher eine Verknüpfung auf die Datenbanktabelle `user_roles` hergestellt, um anhand der Rolle den Typ des Benutzers festzustellen und somit nur typengleiche Benutzer zu zählen. Äquivalent dazu wird in Zeile 12 die Anzahl der Benutzer mit der Variante 1 und einer bestimmten Rolle gezählt und in der Variable «`$var_one_count`» gespeichert.

In Zeile 18 wird der Schwellwert, mit dessen Hilfe die Zuordnung zu Variante 0 oder 1 erfolgt, berechnet und in der Variable «`$scal_threshold`» gespeichert. Die if-Verzweigung in Zeile 20 prüft nun, ob «`$var_one_count`» kleiner als der Schwellwert ist. Sollte dies so sein, wird die Variable «`$var`» auf «1» gesetzt (Zeile 21). Mit dem darauffolgendem SQL-Statement in Zeile 22 wird der Benutzer mit seiner UID, der Test-ID und der Variante mit dem Wert 1 in die Tabelle `user_splitting` eingetragen. Er ist somit der Variante 1 zugeordnet. Sollte der Schwellwert aber größer sein, wird im else-Zweig der if-Verzweigung die Variable «`$var`» auf 0 gesetzt (Zeile 24). Mit dem SQL-Statement in Zeile 25 wird der Benutzer somit der Variante 0 zugeordnet.

Mit den letzten beiden Funktionen des Moduls kann ermittelt werden, in welcher Gruppe der Benutzer ist. Die erste Funktion wird später in den Panels, welche die Benutzerprofile der Au-pair bzw. Gastfamilien darstellen, als Visibility-Rule

verwendet. Genauer dazu ist im nächsten Abschnitt (3.3.2.3) zu finden. Nun soll aber auf die Funktion «user\_splitting\_user\_in\_group» eingegangen werden:

Listing 3.24: Benutzergruppe überprüfen

```

1 function user_splitting_user_in_group($uid, $testid, $group){
2     if (user_splitting_get_group($uid, $testid) == $group){
3         return TRUE;
4     } else {
5         return FALSE;
6     }
7 }

```

Dieser Funktion werden drei Parameter übergeben, nämlich die UID, die Test-ID und eine Group<sup>15</sup> (Zeile 1). Um abzufragen, ob ein Benutzer einer bestimmten Gruppe angehört, muss in der Datenbanktabelle nach ihm gesucht werden. In Zeile 2 wird die Funktion «user\_splitting\_get\_group» aufgerufen, die genau diese Aktion ausführt. Die zurückgelieferten Werte dieser Funktion werden dann in der if-Verzweigung überprüft. Ist der Benutzer in der Group, die überprüft wurde, wird TRUE zurückgegeben (Zeile 3); falls er nicht in der Group ist, ein FALSE (Zeile 5).

Das SQL-Statement, welches abfragt, in welcher Gruppe ein Benutzer ist, wird also in der Funktion «user\_splitting\_get\_group» aufgerufen. Im folgenden Quelltext-Abschnitt wird auf diese eingegangen:

Listing 3.25: Benutzergruppe aus Datenbank auslesen

```

1 function user_splitting_get_group($uid, $testid){
2     $result = db_query("SELECT var FROM {user_splitting} WHERE uid=%d
3         AND test_id=%d", $uid, $testid);
4     return db_result($result);
5 }

```

Das Resultat des SQL-Statements wird in Zeile 2 in der Variable «\$result» gespeichert und in Zeile 3 für die weitere Verwendung zurückgegeben.

Die Entwicklung des «user\_plitting»-Moduls ist somit abgeschlossen. Mit Hilfe dieses Modules sind im Portal A/B-Tests realisierbar. Um den Test zu konfigurieren, stellt das Modul eine Einstellungsmaske zur Verfügung. Hier kann unter anderem

<sup>15</sup>Englisch für Gruppe



bestimmt werden, welche Benutzergruppen und in welchen Teilen die Benutzer in die Testgruppen eingeteilt werden sollen. Neben den für diese Aufgaben notwendigen Methoden stellt das Modul auch Methoden bereit, mit denen es möglich ist, die Gruppenzugehörigkeit der Benutzer abzufragen. Wodurch diese Methoden aufgerufen werden, ist irrelevant. In dieser Arbeit wird lediglich auf eine Methode eingegangen: die Verwendung in Panels.

### 3.3.2.3 Verwendung in Panels

Es wurde bereits erwähnt, dass die beiden letzten Funktionen aus dem Abschnitt 3.3.2.2 in den Benutzerprofil-Panels als Visibility-Rule eingesetzt werden. In diesem Abschnitt wird dieses Vorgehen genauer betrachtet.

«Panels» ist ein Drupal-Modul, welches als Drag&Drop-Content-Manager fungiert. Mittels dieses Panels-Moduls kann man ein Layout erstellen und Content verschiedenster Art in diesem Layout platzieren (vgl. Panels, 2011). Panels wird im Portal für das Layout der Benutzer-Profile verwendet. Diese haben im unteren Bereich in der letzten Zeile<sup>16</sup> des Profiles eine Region namens «Recommendations», wie in Abbildung 3.14 zu sehen ist:

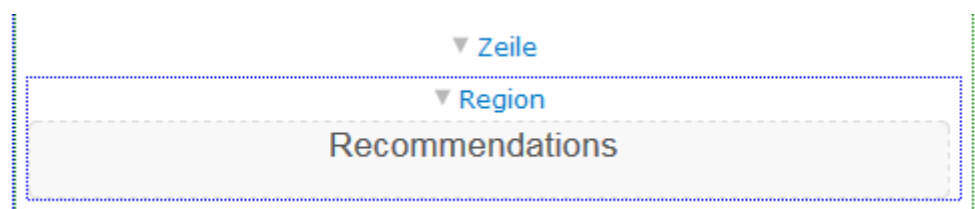


Abbildung 3.14: Recommendations-Region des Panels

In dieser Region wiederum sind die Blöcke eingefügt worden, die nach der Konfiguration des A3S-Moduls zur Verfügung stehen (siehe dazu Abschnitt 3.2.2.1). Die Blöcke zeigen berechnete Empfehlungen an; allerdings mit unterschiedlichen Berechnungsansätzen. Um an dieser Stelle zu differenzieren, dass den Benutzern der

<sup>16</sup>In einem Panel werden horizontal getrennte Bereiche als Zeile bezeichnet.

Variante 0 der «Recommendations-Block 0» und den Benutzern der Variante 1 der «Recommendations-Block 1» angezeigt wird, wurde die Visibility-Rule erstellt. Diese Regel ist in Abbildung 3.15 dargestellt:



Abbildung 3.15: Visibility-Rule des Recommendation-Blocks 0

Der Block wird nur dann angezeigt, wenn der PHP-Code TRUE zurückliefert. Die if-Verzweigung in der Visibility-Rule dient dazu, den Administrator unabhängig von einer Variante zu behandeln, um ihm immer diesen Block anzuzeigen. Sollte es sich bei dem Benutzer, der diesen Block gerade sehen möchte, nicht um den Administrator (also uid != 1) handeln, so wird die Funktion «user\_splitting\_user\_in\_group» aufgerufen. Dieser Funktion werden drei Parameter übergeben, nämlich die UID des Benutzers, die Test-ID und die Group. Der Parameter Group ist mit der Variable «\$var» gleichzusetzen, welche in der Datenbanktabelle gespeichert ist. Liefert diese Funktion TRUE, wird der Block auf dem Profil dargestellt.

Den Benutzern, die sich in einem A/B-Test befinden, werden demzufolge – je nach Gruppenangehörigkeit – unterschiedliche Empfehlungen angezeigt. Für das Auswerten des Testes stellt das Modul «user\_splitting» keine Methoden zur Verfügung. Um den Test auswerten zu können, wird der A3S-Event-Tracking-Service verwendet. Dieser ist in der Lage, das Anklicken von Empfehlungen durch den Benutzer zu verfolgen (siehe dazu Abschnitt 3.2.1). Die dabei entstandenen Daten werden vom A3S-Team für

eine Auswertung bereitgestellt. Auf die Auswertung dieser Daten wird nun genauer eingegangen.

#### 3.3.2.4 Auswertung mittels QlikView

Um ein schnelles und einfaches Auswerten der Daten zu gewährleisten, wurde «QlikView» eingesetzt. „QlikView kombiniert unternehmensweite Analyse- und Suchfunktionalitäten mit der Einfachheit und Anwenderfreundlichkeit, wie man sie von der Office-Umgebung kennt.“ (QlikView, 2010)

Um die Daten auswerten zu können, müssen sie zunächst in QlikView importiert werden. Das QlikView-Projekt, auf das in dieser Arbeit eingegangen wird, importiert die Daten aus einer Backup-Datenbank des Online-Portales. Die Datenbank wurde um eine Tabelle «rcn\_funnel» erweitert, in der die gesammelten Event-Tracking-Daten stehen. Die in dem QlikView-Projekt erstellten Objekte geben einen Überblick über die Anzahl der Benutzer, die einem A/B-Test zugeteilt wurden. Abbildung 3.16 verdeutlicht dies:

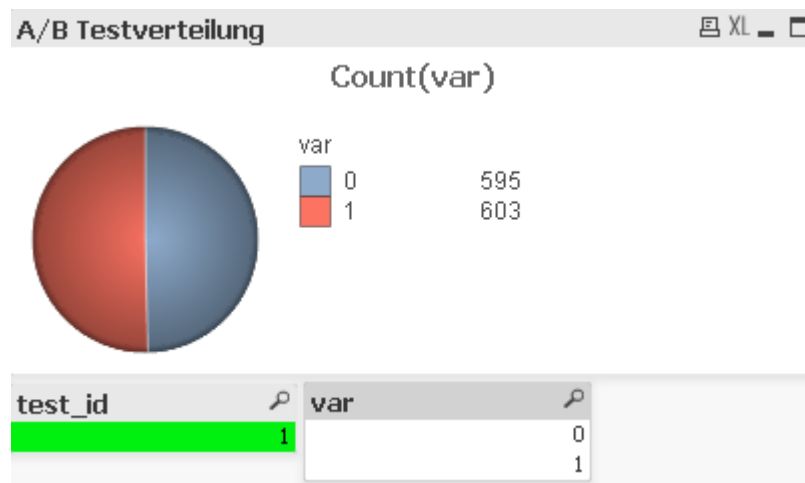


Abbildung 3.16: QlikView: Übersicht über die A/B-Testverteilung

Es ist zu erkennen, dass 595 Benutzer der Variante 0 ( $\text{var} = 0$ ) und 603 Benutzer der Variante 1 ( $\text{var} = 1$ ) an einem A/B-Test mit der «test\_id» 1 teilnehmen. Stellt man nun eine Verbindung zu den angesehenen Profilen der Benutzer her, lassen sich die

Benutzer nach ihrem Typ differenzieren. Indem der «viewed\_type» auf «au\_pair» gesetzt wird, sind in dem Objekt «A/B-Testverteilung» nur noch die ausgewerteten Gastfamilien zu sehen<sup>17</sup>. Die Übersicht über die Gastfamilien des A/B-Testes ist in der Abbildung 3.17 zu sehen:

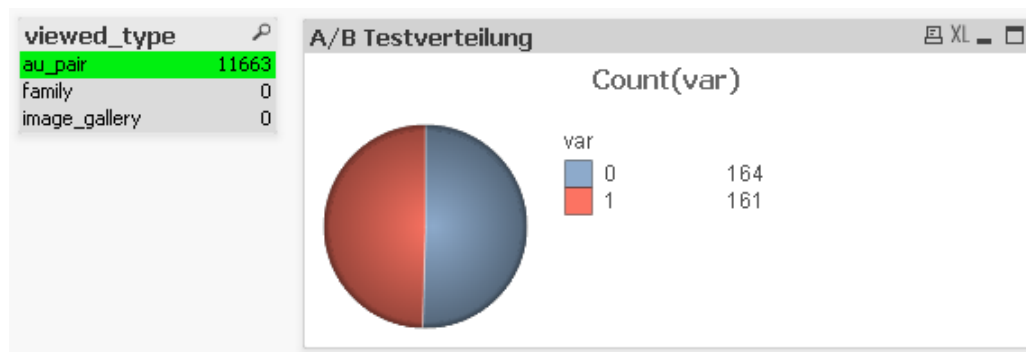


Abbildung 3.17: QlikView: Übersicht über die Gastfamilien des A/B-Testes

Es gehören demnach 164 Gastfamilien der Variante 0 und 161 Gastfamilien der Variante 1 an. Die Zahl hinter «au\_pair» gibt an, dass diese 325 Gastfamilien<sup>18</sup> insgesamt 11663 Au-pair-Profil besucht haben. Um eine Aussage darüber treffen zu können, welche Variante nun zu mehr Profilbesuchen geführt hat, reicht ein Klick auf die «var»-Schaltfläche (rechts vom Kreis) oder auf eine Farbe des Kreises in dem «A/B-Testverteilung[s]»-Objekt aus. Die beiden folgenden Abbildungen zeigen die Ergebnisse dieser Klicks<sup>19</sup>:

<sup>17</sup>Dies ist so möglich, da Gastfamilien im Portal nur Zugriff auf Au-pair-Profil haben. Im Vergleich dazu haben Au-pair nur auf Gastfamilien-Profil Zugriff.

<sup>18</sup>Summe von Variante 0 und Variante 1.

<sup>19</sup>Zur besseren Veranschaulichung wurde der Kreis in Abbildung 3.19 rot gefärbt. Dies ist im QlikView-Projekt leider nicht so; hier wird der Kreis auch blau dargestellt.

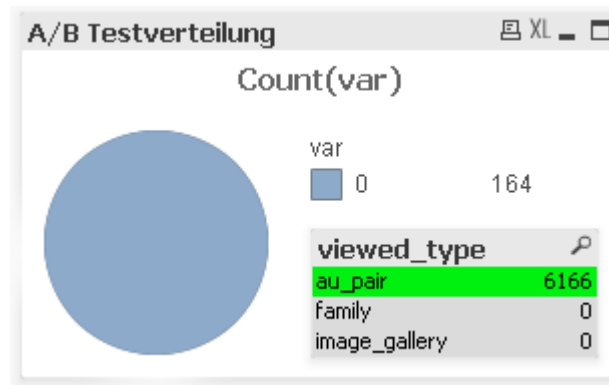


Abbildung 3.18: QlikView: Übersicht über die Variante 0 des A/B-Testes

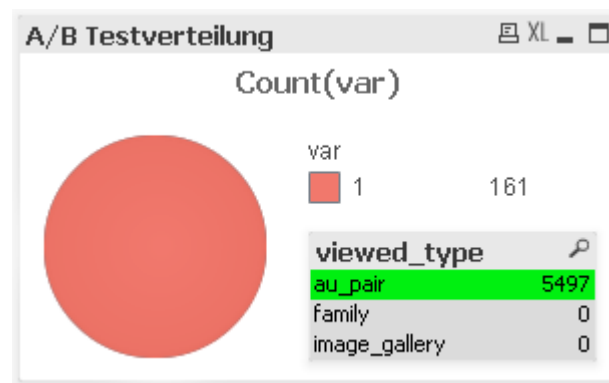


Abbildung 3.19: QlikView: Übersicht über die Variante 1 des A/B-Testes

Nach diesen Bildern zu urteilen, führte die Variante 0 dazu, dass 164 Gastfamilien auf 6166 Au-pair Profile gelangten (Abbildung 3.18). In Variante 1 wurden hingegen nur 5497 Profile durch die 161 Gastfamilien aufgerufen (Abbildung 3.19). Die Variante 0 scheint demnach bessere Empfehlungen zu haben.

Um diese Annahme weiter zu bestärken, wurde dem QlikView-Projekt ein weiteres Objekt hinzugefügt. Das «gesendete Kontaktanfragen»-Objekt verdeutlicht, an wieviel Prozent der aufgerufenen Profile dann auch eine Kontaktanfrage gesendet wurde. Dieser Wert bringt die Qualität der angezeigten Empfehlungen am besten zum Ausdruck, da ein Benutzer, nachdem er zum ersten Mal Interesse an einem anderen Benutzer bekundete, dies noch einmal bestätigt. Somit war die Empfeh-

lung passend für ihn. Die beiden Abbildungen zeigen die Werte des «gesendete[n] Kontaktanfragen»-Objekts in diesem Beispiel:



Abbildung 3.20: QlikView: Übersicht über die gesendeten Kontaktanfragen der Variante 0 des A/B-Testes



Abbildung 3.21: QlikView: Übersicht über die gesendeten Kontaktanfragen der Variante 1 des A/B-Testes

Mittels dieser Abbildungen wurde die Annahme, dass Variante 0 der Empfehlungen besser sei, bestätigt. Diese Variante hat bei 28,32% der dargestellten Profile zu einer Kontaktanfrage geführt (Abbildung 3.20). Variante 1 hatte hingegen nur einen Wert von 13,88% (Abbildung 3.21).

Das Beispiel zeigt, wie einfach es mit QlikView ist, komplexe Daten auszuwerten. Nach einer solchen Auswertung ist es sinnvoll, die bessere Variante zu benutzen und diese eventuell mit einer neuen zu vergleichen. Änderungen an den Berechnungsansätzen können somit verfolgt und gegebenenfalls wieder verworfen werden.

### 3.3.3 Fazit

Abschließend ist zu sagen, dass mit dieser Lösung eine Umgebung geschaffen wurde, um A/B-Tests durchführen zu können. Das Modul trennt Benutzergruppen in

vorgegebene Teile und unterscheidet auch zwischen den Rollen, die die Benutzer haben. Somit können mehrere Benutzergruppen an den Tests teilnehmen. Da die Möglichkeit besteht, mit einer Test-ID die Tests zu nummerieren, können weitere Tests durchgeführt werden.

Das Modul kann auf jedes erdenkliche Element im Portal angewendet werden. Es muss lediglich ein Weg geschaffen werden, über den man die Funktion, die den Benutzer überprüft, ob dieser in der Test-Gruppe ist oder nicht, aufrufen und auswerten kann.

## 4 Schlusswort

In dieser Arbeit wurde auf den Begriff «Usability» und dessen Bedeutung in einem Online-Portal eingegangen. Unter dem Aspekt, es dem Benutzer so einfach wie möglich zu machen, wurden Wege evaluiert, die es ihm ermöglichen sollen, sein Ziel im Portal zufriedenstellend zu erreichen.

Mit der Umsetzung eines «Matchmaking»-Moduls für Drupal wurde eine Lösung geschaffen, individuelle Suchlisten, bei denen ein Benutzer keine Suchfilter bedienen muss, zu generieren. Außerdem ist dieses Modul durch die Anbindung an «Views» in hohem Maße konfigurier- und anpassbar.

Weiterhin ist es gelungen, einen automatisierten Ablauf bereitzustellen, mit dem «Recommendations» – berechnet durch einen Dienstleister – im Portal angezeigt werden können. Durch ein zielgerichtetes Anpassen des Templates dieser Empfehlungen integriert sich diese Anzeige perfekt in das Online-Portal. Das Bereitstellen von Empfehlungen kann durch verschiedene Berechnungsansätze realisiert werden. Um qualitative Unterschiede dieser Berechnungsansätze messen zu können, wurde eine Testumgebung entwickelt und umgesetzt. Diese macht es möglich, verschiedene Empfehlungsarten mittels eines A/B-Testes zu untersuchen. Auswerten lassen sich die Testdaten dann mit dem leistungsstarken Tool «QlikView».

Bei einem Versuch, die Usability eines Systems zu erhöhen, ohne dass eine aussagekräftige empirische Befragung der Benutzer des Systems durchgeführt wird, ist es nötig, sich an Dinge zu halten, die der Benutzer kennt oder kennen könnte.

Festzuhalten ist, dass die geforderten Umsetzungen erfüllt wurden. Leider konnte kein Test des Portales ohne Matchmaking und/oder Recommendations durchgeführt



werden. Deshalb ist es nicht möglich, Aussagen darüber zu treffen, ob sich die Bedienbarkeit des Portales verbessert hat oder nicht. Betrachtet man aber die Anzahl der Benutzer-Profile, die durch Recommendations aufgerufen wurden, ist zumindest davon auszugehen, dass die Einblendung zu regem Interesse führte.

Bei der weiteren Entwicklung des Portales sollte sich an die klassischen «Usability Improvement»-Methoden gehalten werden. Ein Ändern der Linkstruktur, ein eventuelles Eye-Tracking oder das Anpassen des Layouts sind hier aufzuführen.

# A Anhang

## A.1 Tabellen

Typ	Key	Geschlecht
Au-pair	1	weiblich
Au-pair	2	männlich
Gastfamilie	0	egal
Gastfamilie	1	weiblich
Gastfamilie	2	männlich

Tabelle A.1: Key - Wertgegenüberstellung der Geschlechtsangaben

Key	Land	Key	Land	Key	Land
0	Anderes Land	35	Mazedonien	70	Guatemala
1	Ägypten	36	Mexiko	71	Kenia
2	Albanien	37	Moldawien	72	Kirgisistan
3	Argentinien	38	Neuseeland	73	Nepal
4	Armenien	39	Niederlande	74	Mongolei
5	Aserbajdschan	40	Norwegen	75	Bangladesch
6	Australien	41	Österreich	76	Hongkong
7	Belgien	42	Polen	77	Ghana
8	Bosnien-Herzegowina	43	Portugal	78	Südkorea
9	Brasilien	44	Rumänien	79	Nordkorea
10	Bulgarien	45	Russland	80	Madagaskar
11	Chile	46	Schweden	81	Kolumbien
12	China	47	Schweiz	82	Costa Rica
13	Dänemark	48	Singapur	83	Simbabwe
14	Deutschland	49	Slowakei	84	Luxemburg
15	Estland	50	Slowenien	85	Barbados
16	Finnland	51	Spanien	86	Bolivien
17	Frankreich	52	Südafrika	87	Dominikanische Rep.
18	Georgien	53	Tadschikistan	88	Fidschi
19	Griechenland	54	Taiwan	89	Liechtenstein
20	Großbritannien	55	Thailand	90	Malta
21	Indien	56	Tschechien	91	Mauritius
22	Indonesien	57	Tunesien	92	Kamerun
23	Irland	58	Türkei	93	Serbien
24	Island	59	Turkmenistan	94	Uganda
25	Israel	60	Ukraine	95	Venezuela
26	Italien	61	Ungarn		
27	Japan	62	USA		
28	Kanada	63	Usbekistan		
29	Kasachstan	64	VAE		
30	Kosovo	65	Vietnam		
31	Kroatien	66	Weißrussland		
32	Lettland	67	Zypern		
33	Litauen	68	Peru		
34	Marokko	69	Philippinen		

Tabelle A.2: Key - Wertgegenüberstellung der Landesangaben

<b>Feldname im System</b>	<b>Verwendung im System</b>
field_avatar	Benutzerbild
field_gender_search	gesuchtes Geschlecht des Au-pair
field_age	gesuchtes Alter des Au-pair
field_how_much_childs	Anzahl der Kinder der Gastfamilie
field_age_childs	Altersangabe der Kinder der Gastfamilie

Tabelle A.3: Gastfamilien-Felder des Recommendation-Templates und deren Verwendung im System

## A.2 Inhalt der CD

- Diplomarbeit
- HTML-Template
- Module
  - matchmaking
  - rcn\_filter
  - user\_splitting
- QlikView-Projekt

# Glossar

A3S-Host	Der Rechner, der Empfehlungen bereitstellt.
Application Programming Interface	Ist ein Programmteil, der von einem Softwaresystem zur Anbindung an das System zur Verfügung gestellt wird.
Application-Key	Der Application-Key wird benutzt, um die A3S-Services zu differenzieren.
Cascading Style Sheets	Ist eine Formatierungssprache zum Definieren von Style Sheets. Style Sheets sind dabei Stilvorlagen für das Formatieren von strukturierten Dokumenten.
Click Through Rate	Ist die durchschnittliche Anzahl der Clicks bei 100 Einblendungen einer Anzeige und wird in Prozent angegeben.
Comma Separated Values	Ist ein Dateiformat mit einem bestimmten Aufbau. Das Dateiformat wird oft für das Exportieren bzw. Importieren von Datenbankinformationen verwendet.
Content Construction Kit	Ist ein Drupal-Modul, welches es erlaubt, einer Node benutzerdefinierte Felder hinzuzufügen.

Content Management System	Ist eine Art Baukasten, mit dem der Anwender trotz einfacher Computerkenntnisse seine Internet-Präsenz selber erstellen bzw. pflegen kann.
Crypt	Ist eine PHP-Funktion, die unter Verwendung des «DES-basierenden Unix-Standard-Hashing»-Algorithmus´ den übergebenen Wert verschlüsselt.
Drupal	Drupal ist ein Content Management System, welches es erlaubt, verschiedenste Websites ohne Kenntnisse einer Programmiersprache zu erstellen. Es basiert auf einer modularen Architektur, wodurch eine Website benutzerdefiniert funktional erweitert werden kann.
Drupal-Block	Blöcke dienen dazu, zusätzliche Funktionen um den Website Content bereitzustellen. Beispiele dafür sind Login-Block, Sprachwahl-Block oder Recommendation-Block.
Hook	Hooks stellen Schnittstellen bereit, mit denen man sich in bestehende Funktionen hängen kann, um diese zu erweitern oder zu modifizieren.
Host-API	Bezeichner für die vom Host verwendete API-Version. Von dieser ist die Art des Anbindens der eigenen Funktionalität abhängig.
Hypertext Markup Language	Ist eine Auszeichnungssprache, mit welcher der Inhalt von Dokumenten einer Websites strukturiert wird.

Matchen / Matchmaking	Verbinden gleicher Profilangaben von verschiedenen Benutzern.
MD5	Ist eine PHP-Funktion, die unter Verwendung des «RSA Data Security MD5 Message-Digest»-Algorithmus´ den übergebenen Wert verschlüsselt.
Node	Den überwiegenden Teil des Website-Inhalts speichert Drupal in Nodes. Dabei ist es für eine Unterscheidung auch möglich, Nodetypes (Inhaltstypen) anzugeben. Beispiele für Nodes sind Seiten oder Artikel.
Panels	Ist ein Drupal-Modul, welches es ermöglicht, Nodes in Regionen aufzuteilen.
PHP: Hypertext Preprocessor	Ist eine Programmiersprache speziell für Webentwicklungen und lässt sich in HTML einbinden.
Recommendation	Sind Empfehlungen, die dem Benutzer angezeigt werden. Online-Versandhändler setzen diese für eine Umsatzsteigerung ein.
Recommendation-Service	Service eines Dienstleisters, der Empfehlungen berechnet und für die Verwendung bereitstellt.
Structured Query Language	Ist eine Datenbanksprache, mit der entweder Datensätze aus einer Datenbank bereitgestellt oder Änderungen des Datenbankinhalts vorgenommen werden können.

Tooltip	Ist ein Element einer grafischen Benutzeroberfläche und dient als Schnellinformation über einen bestimmten Inhalt. Beispiele dafür sind: Button, Link, Bilder, Schaltflächen, Symbole, etc.
Views	Ist ein Drupal-Modul, mit dem es möglich ist, Listen und Tabellen zu erstellen, zu konfigurieren und anzuzeigen.
Visibility-Rule (Panels)	Mit Hilfe von Visibility-Rules (Sichtbarkeitsregeln) ist es möglich, in Panels den Inhalt der Panels-Regionen nur unter bestimmten Umständen anzuzeigen.



# Literaturverzeichnis

- [psacake 2011] *Pure CSS Tooltip*. Dezember 2011. – URL <http://psacake.com/web/jl.asp>. – letzter Zugriff: 06.01.2012
- [amazon 2011] AMAZON: *amazon*. Dezember 2011. – URL [http://www.amazon.de/Dont-Make-Me-Think-Usability/dp/0321344758/ref=sr\\_1\\_2?ie=UTF8&qid=1324999274&sr=8-2](http://www.amazon.de/Dont-Make-Me-Think-Usability/dp/0321344758/ref=sr_1_2?ie=UTF8&qid=1324999274&sr=8-2). – letzter Zugriff: 06.01.2012
- [Brinkmann 1998] BRINKMANN, Uwe: *htaccess - Paßwortschutz für Web-Bereiche*. August 1998. – URL <http://www.bingo-ev.de/~ub304/htaccess.htm>. – letzter Zugriff: 06.01.2012
- [Burmester u. a. 2001] BURMESTER, Michael ; HASSENZAHL, Marc ; KOLLER, Franz: Usability ist nicht alles - Wege zu attraktiven Produkten. In: *i-com* (2001), Januar, S. 32–40
- [Business-Webhosting.de 2011] BUSINESS-WEBHOSTING.DE: *Cronjob*. Dezember 2011. – URL <http://www.business-webhosting.de/webhosting-lexikon/cronjob.html>. – letzter Zugriff: 06.01.2012
- [Chopra 2010] CHOPRA, Paras: *The Ultimate Guide To A/B Testing*. Juni 2010. – URL <http://www.smashingmagazine.com/2010/06/24/the-ultimate-guide-to-a-b-testing/>. – letzter Zugriff: 06.01.2012
- [Curtis 2011] CURTIS, Dustin: *You should follow me on Twitter*. Dezember 2011. – URL [http://www.dustincurtis.com/you\\_should\\_follow\\_me\\_on\\_twitter.html](http://www.dustincurtis.com/you_should_follow_me_on_twitter.html). – letzter Zugriff: 09.01.2012

- [Dahm 2005] DAHM, Markus: *Grundlagen der Mensch-Computer-Interaktion*. 1. Pearson Studium, Dezember 2005. – ISBN 3827371759
- [Drupal 2011] DRUPAL: *Drupal*. 2011. – URL <http://www.drupalcenter.de/handbuch/infos>. – letzter Zugriff: 06.01.2012
- [Drupalcontrib 2011a] DRUPALCONTRIB: *Content View Field*. 2011. – URL [http://drupalcontrib.org/api/drupal/contributions--views--plugins--views\\_plugin\\_query\\_default.inc/function/views\\_plugin\\_query\\_default%3A%3Aadd\\_where/6](http://drupalcontrib.org/api/drupal/contributions--views--plugins--views_plugin_query_default.inc/function/views_plugin_query_default%3A%3Aadd_where/6). – letzter Zugriff: 06.01.2012
- [Drupalcontrib 2011b] DRUPALCONTRIB: *Hook Views API*. 2011. – URL [http://drupalcontrib.org/api/drupal/contributions--views--docs--docs.php/function/hook\\_views\\_api/6](http://drupalcontrib.org/api/drupal/contributions--views--docs--docs.php/function/hook_views_api/6). – letzter Zugriff: 06.01.2012
- [Drupalcontrib 2011c] DRUPALCONTRIB: *Views Plugin Query Default*. 2011. – URL [http://drupalcontrib.org/api/drupal/contributions--cck--content.module/function/content\\_view\\_field/6](http://drupalcontrib.org/api/drupal/contributions--cck--content.module/function/content_view_field/6). – letzter Zugriff: 06.01.2012
- [Drush 2011] DRUSH: *Drush*. 2011. – URL <http://drupal.org/project/drush>. – letzter Zugriff: 06.01.2012
- [ebay 2011] EBAY: *ebay*. Dezember 2011. – URL [http://www.ebay.de/ctg/Canon-EOS-450D-Rebel-XSi-12-2-MP-DSLR-Kamera-Schwarz-Nur-Gehause-/102202822?\\_pcatid=812&LH\\_ItemCondition=3000#](http://www.ebay.de/ctg/Canon-EOS-450D-Rebel-XSi-12-2-MP-DSLR-Kamera-Schwarz-Nur-Gehause-/102202822?_pcatid=812&LH_ItemCondition=3000#). – letzter Zugriff: 06.01.2012
- [eDarling 2011] EDARLING: *eDarling*. Dezember 2011. – URL <http://www.edarling.de/>. – letzter Zugriff: 06.01.2012
- [Europäisches Komitee für Normung 1995] EUROPÄISCHES KOMITEE FÜR NORMUNG: *Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten*. 1995. – URL [http://www.interactive-quality.de/site/DE/int/pdf/ISO\\_9241-10.pdf](http://www.interactive-quality.de/site/DE/int/pdf/ISO_9241-10.pdf). – letzter Zugriff: 06.01.2012

- [Graf 2008] GRAF, Hagen: *Drupal 6: Websites entwickeln und verwalten mit dem Open Source-CMS*. Addison-Wesley, August 2008. – ISBN 3827326087
- [Gube 2011] GUBE, Jacob: *22 Essential Tools for Testing Your Website's Usability*. <http://mashable.com/2011/09/30/website-usability-tools/>. September 2011. – URL <http://mashable.com/2011/09/30/website-usability-tools/>. – letzter Zugriff: 06.01.2012
- [Kirchhof u. a. 2004] KIRCHHOF, Anja ; GURZKI, Thorsten ; HINDERER, Henning ; VLACHAKIS, Joannis: *Was ist ein Portal? - Definition und Einsatz von Unternehmensportalen*. Juni 2004. – URL <http://www.gurzki.de/wp-content/uploads/2010/10/Whitepaper-Was-ist-ein-Portal-Gurzki3.pdf>. – letzter Zugriff: 06.01.2012
- [Krug 2006] KRUG, Steve: *Don't make me think*. 2. New Riders, Dezember 2006. – ISBN 0321344758
- [Langner 2010] LANGNER, Holger: *A3S Recommendation Services Schritt für Schritt*. Juli 2010
- [Loranger und Nielsen 2008] LORANGER, Hoa ; NIELSEN, Jakob: *Web Usability*. Pearson Deutschland GmbH, Januar 2008. – ISBN 9783827327635
- [marketingterms.com 2011] MARKETINGTERMS.COM: *click-through rate*. Dezember 2011. – URL [http://www.marketingterms.com/dictionary/clickthrough\\_rate/](http://www.marketingterms.com/dictionary/clickthrough_rate/). – letzter Zugriff: 06.01.2012
- [Mayes u. a. 1988] MAYES, J. T. ; DRAPER, Stephen W. ; MCGREGOR, Alison M. ; OATLEY, Keith: Information flow in a user interface: the effect of experience and context on the recall of MacWrite screens. In: *Proceedings of the Fourth Conference of the British Computer Society on People and computers IV*. New York, NY, USA : Cambridge University Press, 1988, S. 275–289. – ISBN 0-521-36553-8
- [Nielsen 1994] NIELSEN, Jakob: *Usability engineering*. Morgan Kaufmann, September 1994. – ISBN 9780125184069

- [Panels 2011] PANELS: *Panels*. 2011. – URL <http://drupal.org/project/panels>. – letzter Zugriff: 06.01.2012
- [QlikView 2010] QLIKVIEW: *Aberdeen-Studie: QlikView-Dashboards übertreffen Mitbewerber in mehreren Bereichen*. September 2010. – URL <http://www.qlikview.com/de/company/press-room/press-releases/2010/de/0913-aberdeen-studie-qlikview-dashboards-bereichen>. – letzter Zugriff: 07.01.2012
- [Rampl 2007] RAMPL, Hansjörg: *Definition Usability*. <http://www.handbuch-usability.de/begriffsdefinition.html>. 2007. – URL <http://www.handbuch-usability.de/begriffsdefinition.html>. – letzter Zugriff: 06.01.2012
- [Richter und Flückiger 2007] RICHTER, Michael ; FLÜCKIGER, Markus D.: *Usability Engineering kompakt: Benutzbare Software gezielt entwickeln*. 1. Spektrum-Akademischer Vlg, Mai 2007. – ISBN 3827418372
- [Robbins 2009] ROBBINS, Jef: *Using Drupal*. O'Reilly Media, Inc., 2009. – ISBN 9780596515805
- [Runte 2000] RUNTE, Matthias: *Personalisierung im Internet*. 1. Deutscher Universitäts-Verlag, 2000. – ISBN 3824405504
- [Schneider 2005] SCHNEIDER, Tobias: *Preference-Based-Recommender-Systeme: individuelle neuronale Präferenzmodellierung am Beispiel von Investmentfonds*. DUV, November 2005. – ISBN 9783824483358
- [SELFHTML 2007] SELFHTML: *SELFHTML: Webserver/CGI / Webserver / .htaccess - Server-Reaktionen kontrollieren*. 2007. – URL <http://de.selfhtml.org/servercgi/server/htaccess.htm>. – letzter Zugriff: 06.01.2012
- [Statistisches Bundesamt 2010] STATISTISCHES BUNDESAMT: *destatis Anteil der Unternehmen mit Online-Verkäufen verdoppelt sich*. November 2010. – URL <http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/>

- Internet/DE/Presse/pm/2010/11/PD10\_\_399\_\_52911.psml. – letzter Zugriff: 06.01.2012
- [Thiemann 2006] THIEMANN, Phillip: *Matchmaking für Web-Services*. Dezember 2006. – URL <http://pvs.uni-muenster.de/pvs/lehre/SS05/seminar/Matchmaking.pdf>. – letzter Zugriff: 06.01.2012
- [Twitter 2009] TWITTER: *Twitter*. Juli 2009. – URL <http://twitter.com/about>. – letzter Zugriff: 06.01.2012
- [Views 2011] VIEWS: *Views*. 2011. – URL <http://drupal.org/project/views>. – letzter Zugriff: 06.01.2012
- [Views Data Export 2011] VIEWS DATA EXPORT: *Views Data Export*. 2011. – URL [http://drupal.org/project/views\\_data\\_export](http://drupal.org/project/views_data_export). – letzter Zugriff: 06.01.2012
- [Völkel 2002] VÖLKEL, Marcus: *Usability als Erfolgsfaktor, Modell zur Usability-Entwicklung und Website-Evaluation*. August 2002. – URL <http://www.scoreberlin.de/fachartikel/usability-score-pyramid/>. – letzter Zugriff: 06.01.2012
- [Williams 2011] WILLIAMS, Steve: *Enterprise Content Management*. Dezember 2011. – URL <http://www.contentmanager.eu.com/portal.htm>. – letzter Zugriff: 06.01.2012

# Erklärung selbstständiger Arbeit

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Quellen stammen, habe ich gekennzeichnet. Die Arbeit hat noch bei keiner anderen Prüfung in gleicher oder ähnlicher Form vorgelegen.

.....  
Bearbeitungsort, Datum

.....  
Unterschrift